POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Vocal tract segmentation of dynamic speech MRI images based on deep learning for neurodegenerative disease application

Tesi di Laurea Magistrale in
Biomedical Engineering - Ingegneria Biomedica

Author: **Angelica Bonà, Matteo Cavicchioli**

Student ID: 945028, 945897
Advisor: Professor Pietro Cerveri
Co-advisors: MSc Matteo Rossi, PhD Maria Luisa Mandelli
Academic Year: 2020-21

Abstract

Motor speech impairments are often the first symptoms in several neurodegenerative diseases. In particular, articulatory errors such as apraxia of speech and/or dysarthria are the most typical initial symptoms of non-fluent/agrammatic variant primary progressive aphasia (nfvPPA), a disorder that progressively debilitates the production of speech. The evaluation of motor speech deficits is still based on perceptual and subjective judgments of clinicians. Dynamic speech MRI (dsMRI) is a non-invasive technique able to image the entire vocal tract and its change over time with high contrast and high temporal resolution, while participants speak in the scanner. The objective of this work is to develop an automatic vocal tract segmentation tool (VTS-tool) by leveraging recent advances in deep learning models (advanced UNets) and dsMRI images to extract the contouring of the main articulators. Specifically, the following articulators were automatically identified: the upper and lower lips, the soft and hard palate, and the tongue with epiglottis. Moreover, we implemented a simple spatial measure to follow the change of articulators over time. We used a dataset composed of 970 dynamic MRI images from 4 young control subjects and 1 nfvPPA patient, representing the mid-sagittal view of their vocal tract during the repetition of specific speech stimuli. First, we provided a manual annotation of the contouring of the main articulators under the supervision of an expert radiologist. These manual segmentations were used to train and test 95 networks composed by the combination of 5 UNets and 19 singular and compound loss functions. Their accuracy was assessed by Dice, Hausdorff Distance and Global Consistency Error metrics. A statistical analysis based on Kruskal Wallis Test was used to identify the three best networks among the 95 tested and a subject-one-out cross validation was conducted to test their generalizability. The best networks showed good metrics results: a median Dice of 0.92, a median Hausdorff Distance of 0.32 and a median Global Consistency Error of 0.0011. Cross validation results also demonstrated that these networks achieve good generalizability and don't suffer from overfitting problem. Best networks were all built with compound loss functions made by three or four losses, proving the superiority of multiple losses compared to double or singular ones.

Keywords: UNet, vocal tract segmentation, compound losses, dynamic speech MRI

Abstract in lingua italiana

I disturbi motori del linguaggio sono spesso i primi sintomi di molte malattie neurodegenerative. In particolare, gli errori articolatori come l'aprassia del linguaggio e/o la disartria sono i sintomi iniziali più tipici dell'afasia progressiva primaria con variante non fluente/agrammatica (nfvPPA), un disturbo che debilita progressivamente la produzione del linguaggio. Ad oggi la valutazione dei deficit motori del linguaggio si basa sulla valutazione percettiva e soggettiva dei clinici. La risonanza magnetica dinamica del parlato (dsMRI) è una tecnica non invasiva in grado di visualizzare l'intero tratto vocale e il suo cambiamento nel tempo durante il parlato, con un contrasto elevato e un'elevata risoluzione temporale. L'obiettivo di questo lavoro è sviluppare uno strumento di segmentazione automatica del tratto vocale (VTS-tool) sfruttando i recenti progressi nei modelli di deep learning (UNet avanzate) e le immagini dsMRI per estrarre il contorno dei principali articolatori. Nello specifico sono stati identificati automaticamente i seguenti articolatori: le labbra superiori e inferiori, il palato molle e duro e la lingua con l'epiglottide. Inoltre, abbiamo implementato una semplice misura spaziale per monitorare l'andamento degli articolatori nel tempo. Abbiamo utilizzato un dataset composto da 970 immagini dsMRI di 4 giovani soggetti di controllo e 1 paziente nfvPPA, che rappresentano la vista medio-sagittale del loro tratto vocale durante la ripetizione di specifici stimoli vocali. In primo luogo, abbiamo fornito un'annotazione manuale del contorno dei principali articolatori sotto la supervisione di un radiologo esperto. Queste segmentazioni manuali sono state utilizzate per addestrare e testare 95 reti formate dalla combinazione di 5 UNet e 19 funzioni di perdita singole e composte. La loro accuratezza è stata valutata dalle metriche Dice, Hausdorff Distance e Global Consistency Error. Un'analisi statistica basata sul test di Kruskal Wallis è stata utilizzata per identificare le tre migliori reti tra le 95 testate ed è stata condotta una subject-one-out cross-validazione per testarne la generalizzabilità. Le migliori reti hanno mostrato buoni risultati: un valore mediano di Dice di 0.92, un valore mediano di Hausdorff Distance di 0.32 e un valore mediano di Global Consistency Error di 0.0011. I risultati della cross-validazione hanno anche dimostrato che queste reti raggiungono una buona generalizzabilità e non presentano problemi di overfitting. Le migliori reti sono state tutte costruite con funzioni di perdita composte formate da tre o quattro elementi, dimostrando la superiorità delle funzioni di perdita multiple rispetto a quelle doppie o singole.

Parole chiave: UNet, segmentazione del tratto vocale, funzioni di perdita composte, risonanza magnetica del parlato

Contents

Introduction

0.1. Clinical Context

Speech articulation is the most complex motor activity humans perform, demanding precise spatio-temporal coordination of more than a hundred muscles. These finely coordinated movements modify the shape of the vocal tract to produce distinct speech sounds. To produce fluent speech, each component needs to be combined with movement of other articulators in an independent fashion [1]. Articulation requires precise motor planning and motor execution and is subserved by a neural network of cortical and subcortical structures [2]. A schematic representation of the anatomy of interest of the vocal tract is depicted in Figure 1. Following the path of the air from outside to inside you meet the upper and lower lips, tongue and hard palate, velum (or soft palate), epiglottis and vocal cords. The dark pink part is the overall vocal tract, which is the portion filled with air.

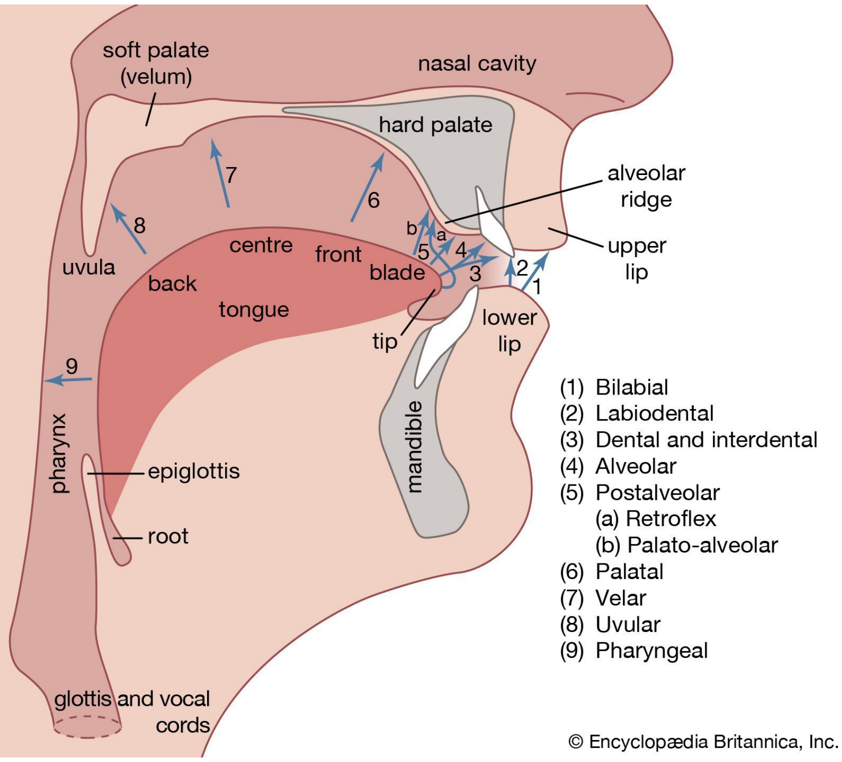


Figure 1: Schematic representation of vocal tract anatomy assessed via `https://www.britannica.com/science/phonetics`

Difficulties with articulation are called motor speech impairments and may be the first symptoms of several neurodegenerative diseases including non-fluent/agrammatic variant primary progressive aphasia (nfvPPA), corticobasal syndrome (CBS), progressive supranuclear palsy (PSP), amyotrophic lateral sclerosis (ALS) and Parkinson's disease [3], [4]. In particular, motor speech impairments are the most common features of nfvPPA, a clinical syndrome most commonly caused by frontotemporal lobar degeneration (FTLD)-tau pathology [5]–[7], manifesting as progressive apraxia of speech (AOS) and/or dysarthria and/or grammatical impairments [8]. In speech production models AOS results from damaged/noisy speech motor program representations, leading to inconsistent speech patterns, consonant and vowel distortions, and alterations to the prosodic and temporal dimensions of speech (motor planning impairment) [9]–[12]. Dysarthria (and its subtypes) lead to a wide range of speech profiles, which include distortions and speech rate impairments, but as a motor execution impairment, occurs in consistent patterns [3]. Motor speech impairments remain difficult to identify and quantify by non-expert clinicians. Currently diagnosis relies on perceptual and subjective judgment of expert Speech-Language Pathologists (SLPs). Automatic acoustic analysis of speech production errors can eventually provide additional objective and quantifiable information if used in tandem with methodologies able to add pathophysiological information. Linking the vocal tract's shape alterations with clinical and acoustic speech evaluations has the potential of better defining the anatomical changes of specific articulation deficits. It also might provide an effective tool for diagnosis and monitoring of neurodegenerative diseases while clarifying and characterizing the origin of speech production errors. However, existing methods to measure or image speech articulation are intrusive and/or cannot provide direct measurements or complete visualization of the dynamics of the entire vocal tract. MRI sequences have several advantages over other existing instrumental approaches that either have limited spatial coverage of the vocal tract (ultrasound, electropalatography), are invasive (cine x-ray and optical coherence tomography), or alter articulatory kinematics (electromagnetic articulography). In particular an innovative MR technology, dynamic speech MRI (dsMRI) offers a unique opportunity for fast, direct, non-invasive, real-time visualization of the changes in the vocal tract, including the deeper articulatory structures, while individuals produce speech in the MRI [13]–[17]. Technical difficulties in implementation, image processing and analysis have so far limited its application to clinical populations [13]–[17]. Recently, advances in the developmental of fast MR imaging sequences have finally helped in significantly improving the temporal resolution of the acquired images using parallel and multiplane imaging techniques [17]–[20].

Despite these important advances, the analysis of these data poses significant challenges mainly in the developing of standardized and automatic post-processing techniques. In general, because of the number of articulators involved in the speech production and the overall complexity of their anatomy, there are still no gold standard datasets or annotations available and this makes difficult to evaluate the performance of proposed automated analysis. The existing techniques used to analyze real time MRI images to investigate speech properties and anatomy can be summarized in four classes as presented in a review paper from Ramanarayanan et al. 2018 [21].

1. *Basic decomposition or matrix factorization techniques*: this class of techniques operates at the whole image level in order to obtain spatio-temporal basis functions of the articulators' movement associated to linguistic gestures;

2. *region of interest (ROI)-based*: this class of techniques is based on the manual demarcation of the regions of interest of which variation can provide useful information regarding linguistic or clinical questions;

3. *grid-based*: this class of techniques is based on a reference coordinate system that is superposed a sagittal view of vocal tract to facilitate the calculation of the vocal

tract area functions by the identification of points of intersection between soft tissue and gridlines;

4. *contour-based*: this class of techniques is based on the extraction of all the tissue boundaries belonging to structures recruited during speech production and provides more detailed anatomical information.

The classification described above is based on the following parameters:

1. processing method;
2. type of output information;
3. auxiliary or prior information necessary to apply the method (e.g. manual labelling, anatomical atlas, prior knowledge);
4. complexity of the implementation;
5. abstraction level of the output information.

Each of this approach presents its advantages and disadvantages but overall the choice of methods and analysis depends on different factors, and most important on the specific research goal. In this thesis, we focus on the last class of methods in order to develop an automated image-segmentation tool to extract the surfaces of the main articulators that are critically involved in speech production. Using this tool, we aim to extract objective and quantitative metrics that can provide useful information to detect motor speech impairments and follow the progression of the disease over time.

0.2. Data Collection

This thesis is integrated into the larger exploratory research study, conducted by a multidisciplinary team from Department of Neurology and Department of Radiology of University California San Francisco (UCSF). This study consists of the enrollment of 10 young and 20 older healthy controls as well as 15 nfvPPA patients from active projects at the UCSF Memory and Aging Center (MAC) and the Language Neurobiology Laboratory (ALBA). Participants are trained first outside of the MR scanner by an SLP who will rehearse with the patient the entire study procedure.

The speech stimuli provided are structured hierarchically to include alternating and sequential diadochokinesis tasks, single syllable and polysyllabic words, and orally read paragraphs. The priority is given to words and stimuli that have been found to elicit errors in patients with neurodegenerative articulatory disorders. This includes high travel words for which the tongue must articulate to the front and then to the back of the mouth (topcop), multiple repetitions of polysyllabic words, and connected speech. By including multiple instances of each consonant, consonant cluster and vowel, a wide range of permissible articulatory movements in Standard American English is recorded. Here below the entire protocol used [22]:

1. diadochokinesis alternating and sequential motion rates (*pa, ta, ka* and *pataka*) as fast and as clearly as possible for 10 seconds;

2. reapeted trials of polysyllabic word repetition: *Microscopic, Segregation, Artillery, Catastrophe, Banana*;

3. repeated trials of fast and clear repetition of a two syllabe word with high articulatory travel: *Topcop*;

4. CVC words to elicit vowels rom across the vowel quadrilateral space: *Heat, Hot, Hurt, Hoot, Hat, Head, Hit, Hub*;

5. grandfather passage reading.

After training, all participants will undergo MRI on a 3T Siemens Prisma scanner equipped with a 64-channel head/neck array where they will repeat the speech stimuli during the MRI acquisition. Pre-recorded audio and video instructions and stimuli will be delivered via headphones and the participants' audio responses will be recorded simultaneously with the MRI acquisition at a sampling frequency of 16 kHz using a FOMRI III optoacoustic fibre-optic microphone with adaptive noise cancellation algorithms. Additional audio post-processing will be performed off-line to reduce the additional scanner noise with denoising algorithms [23]. A mid-sagittal slice for dynamic speech MRI will be acquired during the speech stimuli with a 2D radio-frequency-spoiled radial gradient (RF-spoiled radial GRE) sequence with the following parameters: TR = 2.2 msec, TE = 1.4 msec, flip angle = 4deg, FOV 210x210x10mm3, pixel size = 1.6 x 1.6 mm2, bandwidth = 1500 Hz/pixel. This sequence was implemented in the Department of Radiology at University Medical Center Freiburg [24] and optimized for the 3T scanner at UCSF. The use of higher magnetic fields (3T rather than 1.5T) allows to gain higher signal to noise ratio and better image contrast for segmentation analysis. The acquisition produces approximately 25 frames of a mid-sagittal view per second, or 1500 frames per participant per minute of speaking.

Reconstruction of the images consists of inverse 2D Fourier transformation of the raw data, by using 25 subsequent projections with a sliding-window technique to increase temporal resolution [24]. The reconstruction process will be conducted off-line in Matlab by using virtual machines, with 8 Intel Xeonare (2695 MHz) CPUs.

The study was approved by the UCSF Committee on Human Research and all subjects will provide written informed consent.

0.3. Thesis aim

The purpose of this thesis is to develop an automatic image-segmentation method, exploiting a deep learning approach, to define and extract the contours of the main articulators that contribute to the production of speech sound. This is the first step to better define the anatomical and physiological basis of motor speech errors. The identification of such errors is essential for early differential diagnosis of nfvPPA, when speech-language and pharmacological treatments are most effective, and for disease progression monitoring. For this thesis we used data available from the group of the young healthy subjects to train and test our models and from a nfvPPA patient to test the generalizability of the best models obtained. Unfortunately, because of the restrictions due to the COVID-19 pandemic, only four control subjects and one patient were able to complete the study protocol. Both images and audio have been recorded but, as the scope of the thesis, only images data have been used.

1 | State of the Art

1.1. Dynamic MRI of human speech production

Several techniques are available to investigate the anatomy of vocal tract and its relation to speech production or speech disorders [21], such as X-ray microbeam (XRMB), electropalatography (EPG), electromagnetic articulography (EMA), and ultrasound. The main limitations of these techniques are that they are invasive (the first three), none of them are able to provide a complete view of the vocal tract, and they are not able to provide a high anatomical spatial resolution. On the other hand, magnetic resonance imaging (MRI) is a not invasive technique and it is able to provide a full view of the vocal tract including structural/morphological characteristics of speakers with high spatial resolution. Moreover, with the introduction of fast sequence MRI (dynamic MRI), it has been possible to overcome the limitation of the lower frame rate of classical MRI technique and improve temporal resolution. The limitation of this technique is related to the higher cost, and the reduced accessibility compared the other techniques. Moreover, as the subjects lie supine in the scanner, the effect of gravity due to this position is something that needs to be taken in consideration although previous studies showed that it has minimal effect for speech production analysis. Data collected from RT-MRI have crucial application in linguistic theory, speech modelling and clinical research, as explained in the previous chapter.

In Figure 1.1, we show an example of the vocal tract of a subject acquired with the dynamic MRI sequence. It is a mid-sagittal view of the head and the neck in which each pixel has an intensity of grey that is proportional to the softness of the body tissue represented. Specifically, the softer the tissue the lighter the grey color and vice-versa.

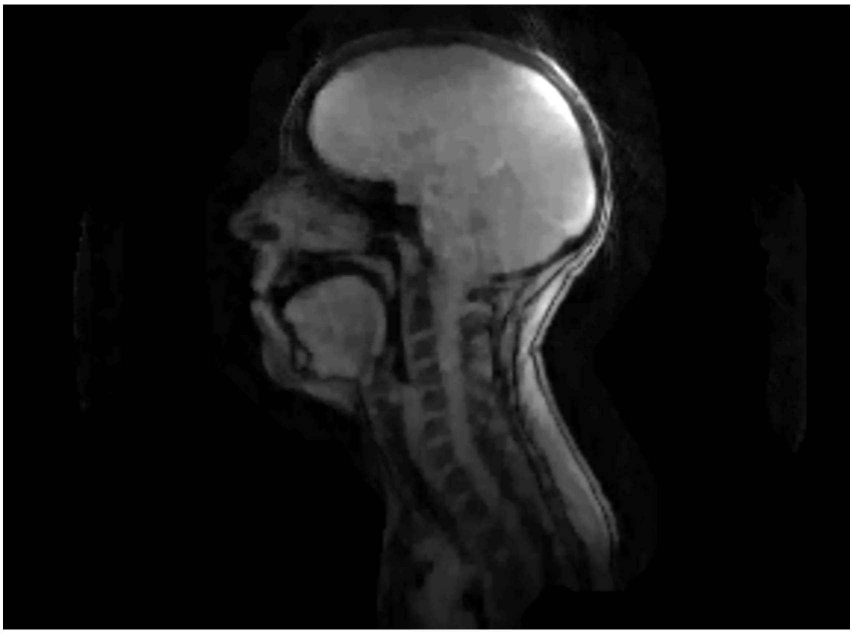


Figure 1.1: Mid-sagittal view of the vocal tract acquired with the dynamic MRI sequence.

1.2. Image segmentation based on deep learning methods

Image segmentation consists of the splitting of an image into disjointed areas according to some features. This means that features are consistent or similar within the same area and different from the others. Particularly, image segmentation belongs to the category of semantic segmentation that is the process of attributing a label to each pixel of an image [25].

Until recently, this task has been addressed using traditional computer vision and machine learning techniques. The introduction of deep learning has brought into the table more accurate and efficient methods to solve these kind of problems. These techniques are useful when dealing with huge amount of data, such as medical images [26]. The management of these data is becoming increasingly possible thanks to improvements in devices capabilities such as GPUs power, memory capacity, power consumption. Compared to traditional segmentation methods, whose segmentation speed was fast but whose accuracy was poor, deep learning methods achieve superior results in terms of accuracy. Moreover, these techniques require less expert analysis since neural networks need only to be trained rather than programmed and they show greater flexibility. Networks can be retrained with different datasets in order to adapt to different use cases. Another important advantage of deep learning techniques compared to traditional methods is the automatic recognition of

underlying patterns and the automatic feature extraction from data. Some disadvantages of deep learning techniques regard the long time necessary to train networks, the power consumption and the necessity to have huge amount of data in order to obtain satisfactory results.
One of the most representative families of deep learning based on artificial neural networks is constituted by Convolutional Neural Networks.

1.2.1. Convolutional Neural Networks

A convolutional neural network (CNN) is an artificial neural network commonly used to analyze visual images. CNNs work by making up of neurons with learnable weights and biases. They are specialized in processing data with a grid-topology such as images, and lean on the linear mathematical operation of convolution, which is generally denoted as follows, in the two-dimensional case [27]:

$$S(m,n) = I(m,n) * K(m,n) = \sum_{i} \sum_{j} I(i,j)K(m-1, n-j), \tag{1.1}$$

The first element is the image input $I(m,n)$ with grid space $[m \times n]$ while the second element is the kernel/filter $K(m,n)$ composed of a series of $[m \times n]$ parameters called weights $w_{ij} \in W$ and bias b. Each kernel is related to a neuron that computes this operation and applies its activation function σ to obtain the output, called feature map and denoted by $S(m,n)$. Each convolutional layer is the basic unit of CNN and is characterized by three fundamental aspects [27]:

- *sparse interactions*: also referred as sparse connectivity or sparse weights, the CNNs kernels have a much smaller number of weights than the input size. This reduces the overall computational cost due to the impossibility to fully connect each input value to each neuron. This type of approach causes each neuron to act on a local region of the input and constitutes a hyper-parameter called receptive field (kernel size) which generally involves $K(m,n) \to K(l,v)$, with $l \ll m$ and $v \ll n$. The kernel is applied to the entire input by sliding it through the input dimensions with a given stride;
- *parameter sharing*: this aspect consists in using the same weights for all kernels that span the input, in order to keep small the total number of parameters and reduce the net complexity;
- *equivariant representation*: the particular form of parameter sharing gives the con-

volution layer the property to change its output in the same way its input changes. $f(x)$ is equivariant to a function $g(x)$ if $f(g(x)) = g(f(x))$.

The weights update of the various kernels, belonging to the network different layers, encloses three steps which are described taking into account a net composed of two layers with respectively (W_1, b_1) and (W_2, b_2) as parameters and activation function σ. I is the input, O is the prediction and Y is the ground truth.

- *forward propagation*: in the first step the input A_0 undergoes to the network configuration to produce the output *out* as depicted in Fig 1.2. At first the input a_0 is convoluted with the weights and bias of the first layer and the product (Z_l) is evaluated by the activation function [27]. This is repeated also for the second layer generating a chain like [28]:

$$\begin{aligned}
&\underline{\text{Forward pass:}}\\
&Z_1 = I * W_1 + b_1\\
&A_1 = \sigma(Z_1)\\
&Z_2 = A_1 * W_2 + b_2\\
&O = \sigma(Z_2)
\end{aligned}$$

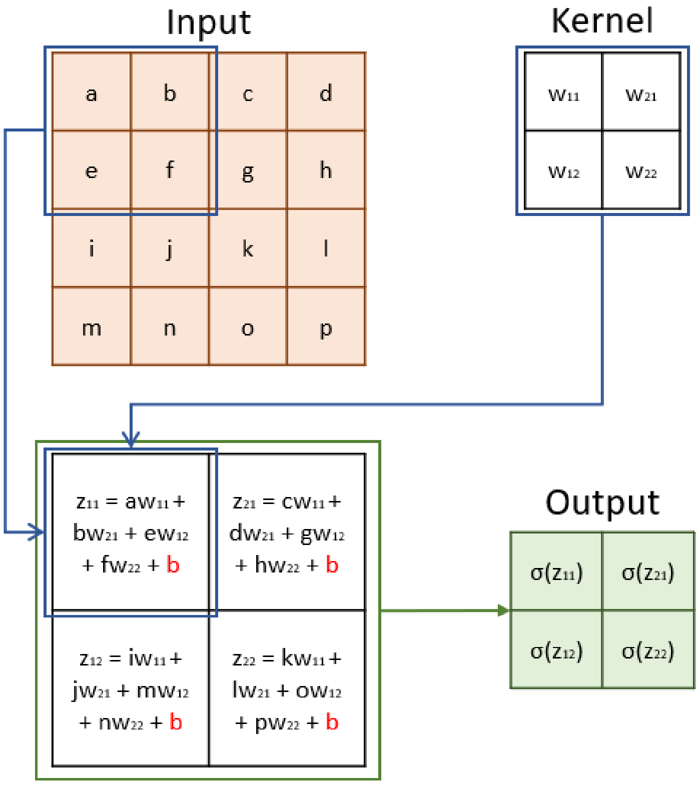

Input
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
Kernel
w_{11}
w_{21}
w_{12}
w_{22}
$z_{11} = aw_{11} + bw_{21} + ew_{12} + fw_{22} + b$
$z_{21} = cw_{11} + dw_{21} + gw_{12} + hw_{22} + b$
$z_{12} = iw_{11} + jw_{21} + mw_{12} + nw_{22} + b$
$z_{22} = kw_{11} + lw_{21} + ow_{12} + pw_{22} + b$
Output
$\sigma(z_{11})$
$\sigma(z_{21})$
$\sigma(z_{12})$
$\sigma(z_{22})$

Figure 1.2: Forward propagation chain of one convolutional layer. Figure adapted from [28]

.

- *loss assessment*: once the net obtains the prediction from the forward propagation, the error between the latter and ground truth must be evaluated to provide the network the directions along which it has to improve. Trying to minimize the loss function L means to find an optimal set of weights and biases such that the prediction O is as close as possible to the ground truth Y, and for doing this the method of **gradient descent** is used. This approach allows to find the global minimum or an acceptable equivalent local minimum of the surface generated from the loss function optimization along the directions provided by the gradient ∇. The loss choice is of crucial importance for the network convergence to the optimal solution and must be accurately performed. During the training of a network, the assessment of a given loss function is performed by the metric, which tells the behaviour of the function during epochs [27], [28];

- *back-propagation*: in the last phase the effective weights and biases update is computed from the output loss function to the first layer. Back-propagation works the other way around, first updates the gradients of the various layers according to the loss gradient of the previous layers and the local gradient, and then proceeds to update the parameters according to the learning rate λ as the following chain:

Gradients update:	Parameters update:
$\nabla_{Z_2} = \nabla_L \cdot \sigma'(Z_2)$	$W_{1_{NEW}} \leftarrow W_1 - \lambda \cdot \nabla_{W_1}$
$\nabla_{b_2} = \nabla_{Z_2}$	$b_{1_{NEW}} \leftarrow b_1 - \lambda \cdot \nabla_{b_1}$
$\nabla_{W_2} = A_1^T \cdot \nabla Z_2$	$W_{2_{NEW}} \leftarrow W_2 - \lambda \cdot \nabla_{W_2}$
$\nabla_{Z_1} = \nabla_{A_1} \cdot \sigma'(Z_1)$	$b_{2_{NEW}} \leftarrow b_2 - \lambda \cdot \nabla_{b_2}$
$\nabla_{b_1} = \nabla_{Z_1}$	
$\nabla_{W_1} = I^T \cdot \nabla Z_1$	

In the particular case of CNNs, the back-propagation can be seen as a convolution with spatially-flipped filters. Moreover the optimization suffers from the gradient vanishing/exploding problem which can limit/exaggerate learning [27], [28].

This three steps are continuously repeated until the network converges to the optimal solution and the combination between proper loss function and framework greatly affects the final performance.

Summarizing, CNNs are preferable with respect to Feed Forward Neural Networks because the full connectivity of the latter ones produces a huge amount of parameters to be learnt. As a consequence, the learning process is greatly slowed down and the network risks to overfit. Moreover, since CNNs have only the *encoding path* that progressively reduces the dimension of the input, they are mainly used for classification tasks rather than for segmentation tasks.

UNet

For segmentation tasks the most used networks are UNet, which are u-shaped CNN made by basically three components [29]. The first is the *encoding path*, where the net encompasses a deeper *analysis* of the topological space with a high receptive field; here the input grid-size is down-sampled and dilated to multiple feature maps. Then the *bottleneck* of the net tries to manipulate this information for a first general learning, allowing *decoding path* to efficiently *synthesize* the knowledge into the output classes. In the UNet generally the *encoding path* and *bottleneck* employ convolution, while *decoding path* uses deconvolution (transpose-convolution). This last operation learns the parameters in the same way as convolutional levels, but its final task is to obtain a result that is topologically the most similar to the up-sampled parallel map, and report the information necessary to complete the objective, reducing the output; here the output is up-sampled and compressed to fewer feature maps. The presence of the *decoding path* provokes a significant increase of the number of parameters to be updated and the gradient vanishing problem. To overcome this problem skip connections between encoding and decoding layers are introduced. UNet are specifically employed for segmentation tasks due to the presence of the deconvolution path that performs upsampling, allowing to obtain an output with the same dimensions of the input. Eventually, it is necessary to find a trade-off of the network depth because the greatest is the depth the more information is extracted from the input but the higher is the overfitting risk.

A couple of applications of UNet on vocal tract segmentation task is reported below.

Erattakulangara et al. proposed a method based on identifying the entire surface covered by air [30]. They fed the network with MRI images and the corresponding manual segmentations, which constitute the ground truth images. In Figure 1.3 , adapted from [30], it's possible to see the input of the network, which is a mid-sagittal MRI image, and the output, which is the airway segmentation (represented as a binary mask). As can be clearly seen, the network used is a U-Net because it has an encoder-decoder structure together with inter-layer connections.

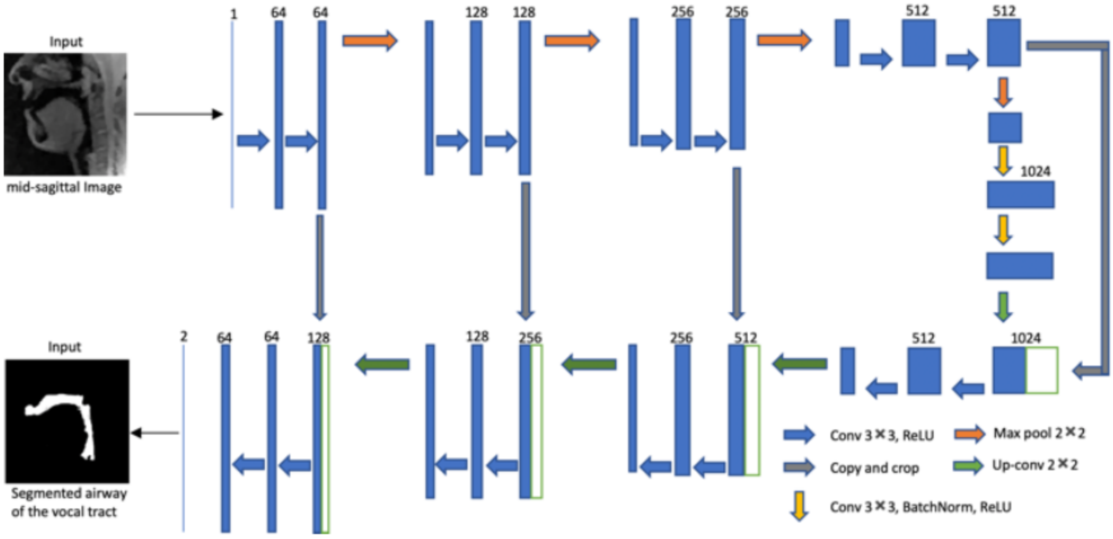


Figure 1.3: Representation of UNet architecture adapted from [30]

In this work 100 MRI images from 10 different subjects were used to train, validate and test the network. During training a binary cross-entropy loss function was used, while the metric adopted was Dice similarity coefficient. This method provided quite similar results compared to the manual segmentations.

The other work was the one of Ruthven et al. that proposed a supervised method to obtain a complete segmentation of all the main articulators, in order to analyse their shape, size, motion and position in a more accurate manner [31]. The authors conducted their study using 392 MRI images from 5 different subjects. The ground truth segmentations were obtained manually by performing a pixel-wise labelling of all the images. They labelled 6 regions, thus obtaining 6 classes: head (including upper lip and hard palate), soft palate, jaw (including lower lip), tongue (including epiglottis), vocal tract and lower incisor tooth space. The authors built a network similar to the original UNet (Figure 1.4) and trained it using a cross-entropy loss function that was weighted in order to compensate for class imbalance. Moreover, they used two metrics to evaluate the performance of the network: Dice similarity coefficient and the Hausdorff distance.

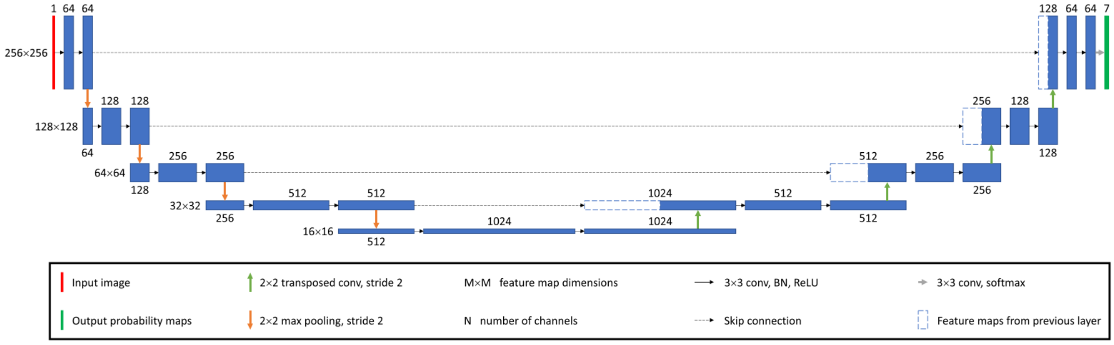


Figure 1.4: Representation of UNet architecture adapted from [31]

The two metrics obtained good results when comparing the automatic segmentation provided by the network and the manual one. They also tested the network on different images (not included in the considered dataset) obtaining satisfactory results. This means that the network achieved a good generalizability. This work constitutes the starting point for our thesis project.

2 | Materials and Methods

This chapter is organized in three main stages:

1. the first stage consists in the description of the main steps needed to obtain and organize the dataset;
2. the second stage consists in the description of the core elements of a neural network: metrics, losses and architectures together with the procedure used to select, train and evaluate networks that are built to accomplish the automatic segmentation on the aforementioned dataset;
3. the third stage consists in post-processing of the segmentations obtained and the description of the Vocal Tract Segmentation tool (VTS-tool).

2.1. Dataset realization

2.1.1. Data collection in video format and protocol description

As mentioned in Chapter Introduction, the preliminary stage of the cross-sectional study that will be conducted by a multidisciplinary team from Department of Neurology and Department of Radiology of University California San Francisco (UCSF), implies the recruitment of 10 young and 20 older healthy subjects (controls) to evaluate its feasibility. But, due to pandemic issues, data from only 4 subjects of the young group were actually collected. Moreover, the first two subjects underwent the entire protocol while the other two only a portion of it. Also the only patient analysed underwent only a portion of the entire protocol.

Here below the protocol version used in clinical practise during acquisitions:

1. repeat PA for 10 sec;
2. repeat TA for 10 sec;
3. repeat KA for 10 sec;
4. repeat PATAKA for 10 sec;

5. subject asked to SWALLOW for 10 sec;
6. repeat MICROSCOPIC for 10 sec;
7. repeat SEGREGATION for 10 sec;
8. repeat ARTILLERY for 10 sec;
9. repeat CATASTROPHE for 10 sec;
10. repeat BANANA for 10 sec;
11. repeat TOPCOP for 10 sec;
12. repeat WELCOME for 10 sec;
13. COUNT for 10 sec.

Each point of the protocol is referred to as task. An acquisition consists in a subject performing only one of these. Each acquisition was conducted using the 3T Siemens Prisma scanner and it produced approximately 25 MRI frames of a mid-sagittal view per second of speaking. Since tasks last 10 seconds, each acquisition contains a total number of frames equal to 250, grouped in a single video (extension .avi). Actually each video lasts about 14 seconds so the maximum number of frames available is 354.

Three tasks among the available ones were chosen, for each subject, trying to guarantee the greatest possible variability in vocal tract motions. This way, 12 videos were considered to train and test the networks while 3 videos, coming from the patient, were used to evaluate the generalizability of the best networks. The 15 considered videos can be seen in Table 2.1, where it's reported the Subject ID, the task, their attendance to the entire protocol or not and the belonging to patient or control group.

Table 2.1: List of videos used for the generation of the dataset. The 'Entire protocol' column highlights whether the complete tasks protocol is available for the subject identified in the 'Subject ID' column.

Subject ID	Task	Entire protocol	Control/Patient
1	SEGREGATION	YES	Control
1	MICROSCOPIC	YES	Control
1	TOPCOP	YES	Control
2	MICROSCOPIC	YES	Control
2	SEGREGATION	YES	Control
2	TOPCOP	YES	Control
3	PATAKA	NO	Control
3	MICROSCOPIC	NO	Control
3	WELCOME	NO	Control
4	PA	NO	Control
4	KA	NO	Control
4	COUNT	NO	Control
5	SEGREGATION	NO	Patient
5	MICROSCOPIC	NO	Patient
5	TOPCOP	NO	Patient

2.1.2. GUI realization and frames extraction

A simple graphical user interface (GUI) was developed in Python 3.7.9 to extract the desired number of frames from videos and to create and organize the dataset. From the interface it's possible to select Dataset directory, the video and the desired number of images until a maximum of 354 (given by the temporal resolution in acquisition phase). Images are extracted equally spaced such that the entire variability of vocal tract movements can be captured. Then they are saved in the specially created folder *Images* in correspondence of the appropriate sub-folder, reporting the task name. Other two folders are created simultaneously with the folder *Images*: *Video*, containing the video and *Segmentations*, which will be filled with the corresponding manual segmentations. These three folders are all included in the parent folder which indicates the subject. A schematic representation of the dataset organization described is depicted in Figure 2.1.

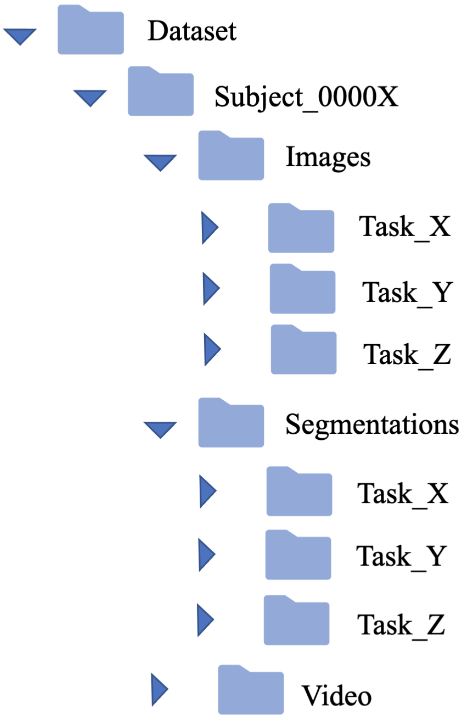


Figure 2.1: Dataset organization; each subject belongs to a folder with its own identification code, which contains the videos, the video frames and the manual segmentations performed on them, divided by the tasks performed by the subject.

Moreover, the interface informs the user about:

- the selection of a new subject with respect to the ones already present;
- the selection of a new task of a subject already present;
- the selection of the same subject and task giving the possibility to increment the number of frames previously extracted, always in equally spaced manner.

The number of frames extracted from videos is different among the five subjects, specifically: 280 frames for subject 1, 240 frames for subject 2, 150 for subject 3, 150 for subject 4, 150 for subject 5. So the total amount of frames used to train and test the networks is equal to 820, while 150 frames were used to evaluate their generalizability.
Images are in *.tiff* format because their quality is the prior characteristic that must be preserved. They are named with the same rules of videos but with the addition of the number of the image itself (e.g. s00001_tSEGREGATION_n01_f61.tiff, where s stands for subject, t for task, n for number and f for frame). An example of an image is provided in Figure 2.2. As can be seen it is a mid-sagittal view of the vocal tract and it consists

of a set of pixels distributed within a grid of dimensions 256 × 256. Each pixel has values between 0 (black colour) and 256 (white colour).

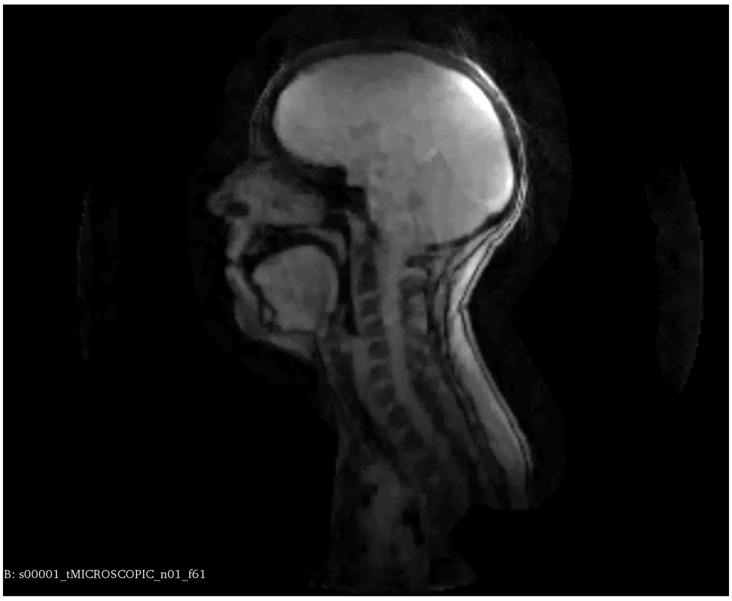


Figure 2.2: Example of MRI image extracted from video.

2.1.3. Manual segmentation of images

All images were segmented manually under the supervision of an expert radiologist that solved some critical issues. He also segmented directly some slots of images. The manual segmentation was performed using 3D Slicer to obtain the ground truth segmentations and it took approximately 15 minutes for each image. Taking into account that the total number of images to be segmented were 970 and the onset of fatigue of the operators after about 20/25 images, it took approximately 250 hours to complete the segmentation process. As can be seen in Figure 2.3, 7 regions of interest were identified:

1. Upper Lip (UL) coloured in green;
2. Hard Palate (HP) coloured in yellow;
3. Soft Palate (SP) coloured in soft brown;
4. Tongue and Epiglottis (TO) coloured in light blue;
5. Lower Lip and Jaw (LL) coloured in red;
6. Head (HE) coloured in orange;
7. Background (BK) that is the whole remaining region coloured in black or dark grey.

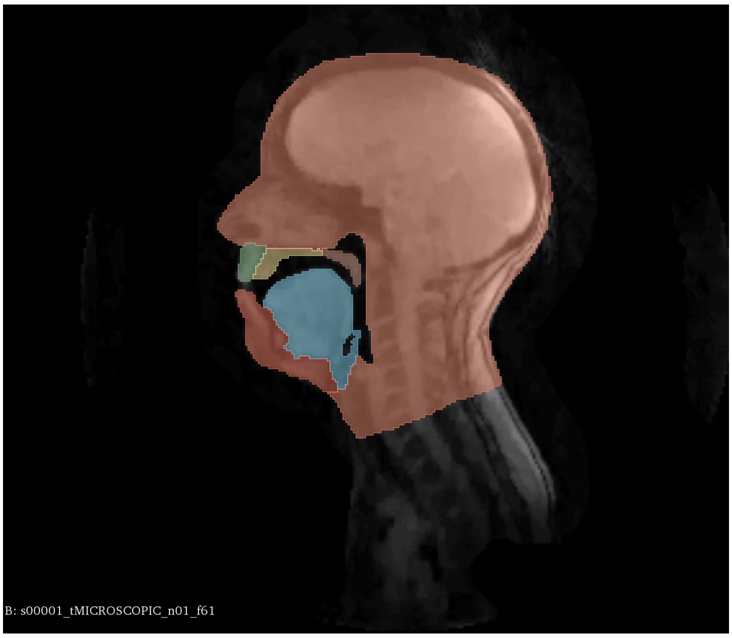


Figure 2.3: Example of a manually segmented image, with the 7 visible regions

The ground truth segmentations have the same dimensions of the images and were saved with their own name but in *.nrrd* format (e.g. s00001_tSEGREGATION_n01_f61.nrrd).

2.2. Dataset preparation

Since manual segmentation was performed by 5 different operators, the correspondence between *label name* (e.g. UL) and *label value* (e.g. 1) needed to be checked and, if necessary, corrected. Specifically, *label name* is the one assigned by the user during manual segmentation, *label value* is a value between 1 and 7 that is assigned automatically to each label, by 3D Slicer, according to the order followed during segmentation. The correspondence established a priori is: UL identified by 1, HP by 2, SP by 3, TO by 4, LL by 5, HE by 6, BK by 7.

Since images pixels assumed values between 0 and 256, they needed to be normalized between 0 and 1 to obtain better performances. A min-max normalization was applied using 0 as minimum value and the 90th percentile (about 130) as maximum value. This decision was taken because only few pixels overcame that value and it was possible to obtain slightly larger values of the normalized pixels. This normalization was performed by Medical Images Dataloader, which is a general-purpose dataloader for Tensorflow 2.x that supports many medical image formats (https://github.com/mrossi93/med_dataloader). Checked segmentations and images were processed by dataloader that allows to split the dataset into *Training*, *Validation* and *Test* sets according to some percentages that were established to be 80%, 10% and 10% respectively.

2.2.1. Dataset visualization

In Figure 2.4 can be seen the 7 classes separately, infact the Neural Network will produce 7 predictions starting from one single image, one for each class.

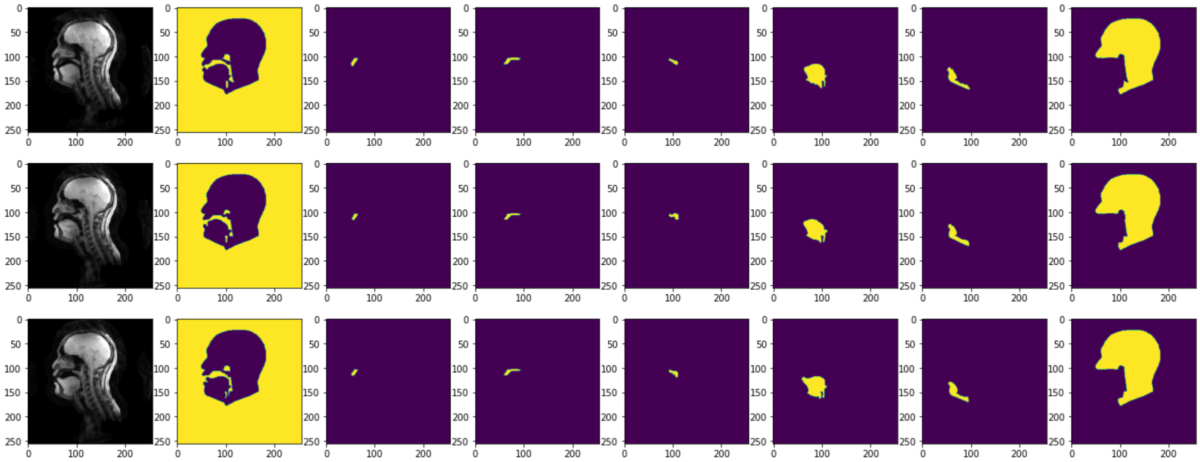


Figure 2.4: Manual segmented classes visualization. From left to right: Background, Upper lip, Hard palate, Soft palate, Tongue and epiglottis, Lower lip and jaw, Head

Manual segmentations are defined as *crisp*, which means that their pixels can assume either 0 value (purple) or 1 value (yellow). Segmentations predicted by the Neural Network are instead defined as *fuzzy* because their pixels can assume any value between 0 and 1. Moreover, non zero pixels belong to the so called *foreground*, whereas zero pixels belong to the so called *background*.

2.3. Metrics

Assessing the accuracy and quality of segmentation algorithms is of great importance; segmentation performance is carried out by the metric, which compares the images predicted by the network (predictions) to the manually segmented images (labels), considered as ground truth.
Metrics should have specific requirements:

- *accuracy*: it is affected by delineation of the contours and segmented dimension;
- *sensitivity*: it refers to all those aspects that can destabilize it, such as outliers, the number of segmented objects, the imbalance of classes (number of pixels of the

foreground are imbalanced with respect to the pixels of background);

- *efficiency*: it is a concept closely related to the processing time that is often a limit and does not allow a theoretically functioning metric to function at the computational level.

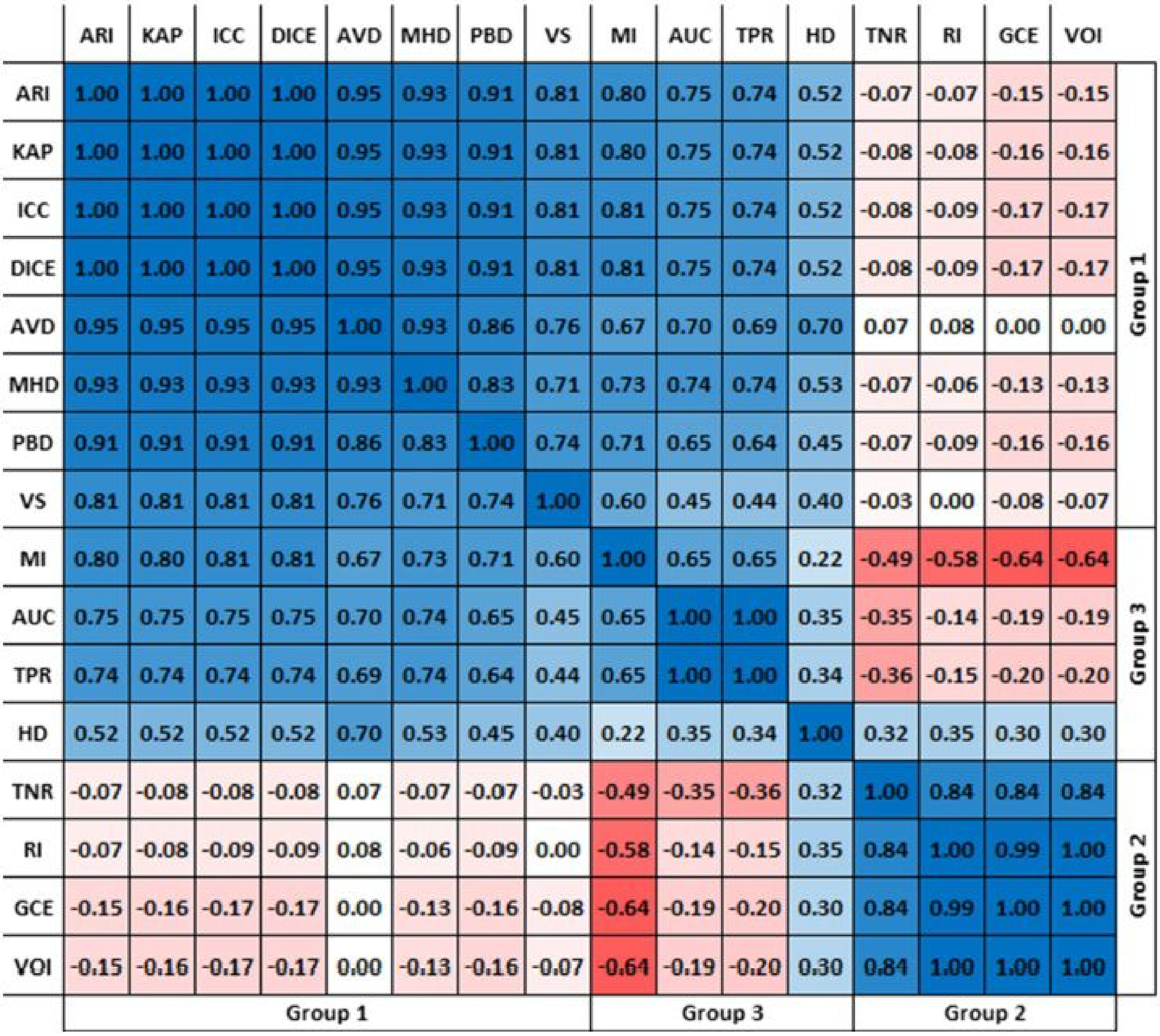
Different metrics have different operating methods and focus on different aspects: some of them focus more on accuracy aspects than others, for example. They are also affected differently by the aspects of sensitivity. For these reasons metrics to be used must be carefully selected. To correctly choose the metrics, the guidelines provided by [32] were used, as depicted in Table 2.2, adapted from [32]. In this table there is a synthesis of some segmentation properties, explained below, and the corresponding recommended or not recommended metrics.

- *Outliers exist*: they are small points or regions segmented outside the correct area;
- *small segment*: foreground area is significantly smaller than the background one;
- *complex boundary*: some segments have a non-regular shaped complex boundary;
- *low densities*: predictions are not solid areas but have some little empty spaces;
- *low segmentation quality*: predictions have low overlap with the ground truth;
- *contour is important*: contour of segmentation is of high importance;
- *alignment is important*: alignment between prediction and segmentation is more important than the contour;
- *recall is important*: necessity to include all the ground truth area in the prediction, regardless of borders and possible false positives;
- *volume is important*: magnitude of segmented region is more important than boundary and alignment;
- *general shape & alignment*: general shape and alignment is more significant than the exact shape and alignment.

Properties that are most inherent to the considered task are: Small segment, Complex boundary, Low densities, Contour is important. Then, among the metrics that meet the necessary criteria, the most uncorrelated ones were identified, consulting the correlation matrix presented in the Figure 2.5, adapted from [32], so as to quantify in a complementary way the goodness of the segmentations.

Table 2.2: In this table, adapted from [32], rows represent the requirements and properties, while columns represent the metrics. Cell with (X) means that the metric has a bad performance for the requirement/property, while (✓) means well performance and empty cell means neutrality. Requirements/Properties and the metrics selected for this work are in bold.

	DICE	JAC	TPR	TNR	FPR	FNR	FMS	VS	**GCE**	RI	ARI	MI	VOI	ICC	PBD	KAP	AUC	**HD**	AVD	MHD
Outliers exist	✓	✓					✓	✓				✓	✓			✓	✓	X	✓	✓
Small segment	X	X	X	X	X	X	X			X	X	X	X			X	X	✓	✓	✓
Complex boundary								X										✓	✓	X
Low densities	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	✓	✓	✓
Low segmentation quality								X										✓	✓	✓
Contour is important								X										✓	✓	X
Alignment is important								X												
Recall is important			✓									✓								
Volume is important								✓												
General shape & alignment	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	✓



	ARI	KAP	ICC	DICE	AVD	MHD	PBD	VS	MI	AUC	TPR	HD	TNR	RI	GCE	VOI	
ARI	1.00	1.00	1.00	1.00	0.95	0.93	0.91	0.81	0.80	0.75	0.74	0.52	-0.07	-0.07	-0.15	-0.15	Group 1
KAP	1.00	1.00	1.00	1.00	0.95	0.93	0.91	0.81	0.80	0.75	0.74	0.52	-0.08	-0.08	-0.16	-0.16	
ICC	1.00	1.00	1.00	1.00	0.95	0.93	0.91	0.81	0.81	0.75	0.74	0.52	-0.08	-0.09	-0.17	-0.17	
DICE	1.00	1.00	1.00	1.00	0.95	0.93	0.91	0.81	0.81	0.75	0.74	0.52	-0.08	-0.09	-0.17	-0.17	
AVD	0.95	0.95	0.95	0.95	1.00	0.93	0.86	0.76	0.67	0.70	0.69	0.70	0.07	0.08	0.00	0.00	
MHD	0.93	0.93	0.93	0.93	0.93	1.00	0.83	0.71	0.73	0.74	0.74	0.53	-0.07	-0.06	-0.13	-0.13	
PBD	0.91	0.91	0.91	0.91	0.86	0.83	1.00	0.74	0.71	0.65	0.64	0.45	-0.07	-0.09	-0.16	-0.16	
VS	0.81	0.81	0.81	0.81	0.76	0.71	0.74	1.00	0.60	0.45	0.44	0.40	-0.03	0.00	-0.08	-0.07	
MI	0.80	0.80	0.81	0.81	0.67	0.73	0.71	0.60	1.00	0.65	0.65	0.22	-0.49	-0.58	-0.64	-0.64	Group 3
AUC	0.75	0.75	0.75	0.75	0.70	0.74	0.65	0.45	0.65	1.00	1.00	0.35	-0.35	-0.14	-0.19	-0.19	
TPR	0.74	0.74	0.74	0.74	0.69	0.74	0.64	0.44	0.65	1.00	1.00	0.34	-0.36	-0.15	-0.20	-0.20	
HD	0.52	0.52	0.52	0.52	0.70	0.53	0.45	0.40	0.22	0.35	0.34	1.00	0.32	0.35	0.30	0.30	
TNR	-0.07	-0.08	-0.08	-0.08	0.07	-0.07	-0.07	-0.03	-0.49	-0.35	-0.36	0.32	1.00	0.84	0.84	0.84	Group 2
RI	-0.07	-0.08	-0.09	-0.09	0.08	-0.06	-0.09	0.00	-0.58	-0.14	-0.15	0.35	0.84	1.00	0.99	1.00	
GCE	-0.15	-0.16	-0.17	-0.17	0.00	-0.13	-0.16	-0.08	-0.64	-0.19	-0.20	0.30	0.84	0.99	1.00	1.00	
VOI	-0.15	-0.16	-0.17	-0.17	0.00	-0.13	-0.16	-0.07	-0.64	-0.19	-0.20	0.30	0.84	1.00	1.00	1.00	
	Group 1								Group 3				Group 2				

Figure 2.5: Correlation between metrics adapted from [32]. Red cell stands for inverse correlation, blue cell for direct correlation. The strength of correlation is denoted by the colour intensity and the cell value.

Following these considerations, Hausdorff Distance (HD), Dice coefficient (DICE) and Global Consistency Error (GCE) were chosen as appropriate metrics.

2.3.1. Hausdorff Distance

The Hausdorff Distance (HD) belongs to the class of spatial distance based metrics and it's a min-max distance between two sets of points. Its dissimilarity measurement takes into account the position of each point in each set and, for this reason, it is used to quantify the precision of the prediction's spatial position and boundaries with respect to the ground truth; due to this approach, HD suffers a lot from outliers and above all it has a high computational cost, which in most cases leads necessarily to its indirect calculation, in order to preserve the efficiency. To better explain the Hausdorff Distance the definition of directed HD (dHD), evaluated by a set of points G (belonging to the ground truth) towards a set of points P (belonging to the prediction), is mandatory and denoted by $h_{G \to P}(G, P)$. Considering that $g \in G$ and $p \in P$, in the primary step all the distances between g_1 and all the points p_i of P must be calculated and the minimum selected (nearest neighbor). This procedure must be repeated for all points g_i of G and once all the minimum distances were obtained, the greater is the directed HD. As regards the directed HD from P to G, $h_{P \to G}(P, G)$, the process takes place in the same way, but it's important to note that the two dHDs are not symmetrical, as shown in Equation (2.1a) and Equation (2.1b) .

$$\begin{cases} h_{G \to P}(G, P) = \max_{g \in G} \min_{p \in P} ||g - p||, & (2.1a) \\ h_{P \to G}(P, G) = \max_{p \in P} \min_{g \in G} ||p - g||, & (2.1b) \end{cases}$$

The overall HD is instead symmetric and obtained from the maximum between (2.1a) and (2.1b), as shown in Equation (2.2).

$$HD(G, P) = max(h(G, P), h(P, G)), \tag{2.2}$$

As regards image segmentation, points and sets of the formula are respectively pixels and images. As mentioned before, the direct calculation of HD is not very efficient due to these reasons:

- point set size (pixels/voxels belonging to the segmented area/volume): as long as it can reach very high values in some cases (e.g. MRI of the whole body includes images with 10 million pixels), it can lead to unreasonable runtimes;
- grid size (whole pixels/voxel including background): as long as it can be much greater than the size of the point set, it adds to the processing a considerable number of pixels that could be avoided;
- sparsity and Density of the points: they can affect in a different manner but a sparse set is preferable to a dense one;
- generality: the algorithm used to compute HD must be able to deal with all use cases.

Taha et al.[33] proposed an optimized algorithm called Early Break Technique and developed by SciPy. It considers all the aspects listed above to evaluate the dHD. For the scope of this thesis a computation function based on the SciPy framework was therefore implemented and adapted to work with rank 3 tensors [34].

The metric was expressed in HD unit [HDu] which, in real space, can assume any value between 0 (ideal prediction) and ∞ (very bad prediction). However, considering the space that can be occupied by two objects, the contribution to dHD is limited to the grid space. It was possible to find a maximum dHD to normalize the [HDu] in a range between [0,1]. To do this, the set of starting points belonging to the ground truth (G), and all the points of background (B) were considered and, by means of the distance transform map (dtm), each pixel of the set was labelled with the distance value from the nearest null pixel, using the Euclidean distance as distance metric. Once the map was obtained, the maximum distance value, which represents the furthest point b_{fg} from the set G, was selected. Then the dHD between the set and the furthest point was evaluated as in (2.3a). The operation had to be repeated also considering P as the starting set, evaluating b_{fp} with (2.3b) and, once both the most distant points were found, the maximum had to be chosen to normalize the overall HD.

This last procedure was feasible on a theoretical level, but not on a practical level, entailing a high computational cost to be added to the one already present. To overcome this problem, the evaluation of the maximum point was made in the pre-processing phase, limiting it only to the set knowable a-priori, that is the ground truth. Furthermore, since the dataset is made up of several segmentations, the average space occupation of all the labels of the same type was used as a set of G points, in order to have a set of b_{fg} points for all the labels equal to the number of classes that make up the segmentation. The final normalization is shown in Equation (2.4). This procedure leads to an overestimation of the error, which however is acceptable.

$$\begin{cases} b_{fg} = max\, h_{G \to b_{fg}}(G, b_{fg}), & (2.3a) \\ b_{fp} = max\, h_{P \to b_{fp}}(P, b_{fp}), & (2.3b) \end{cases}$$

$$||HD(G,P)|| = \frac{HD(G,P)}{b_{fg}}, \tag{2.4}$$

Furthermore, HD does not suffer from the class imbalance problem, being a distance measure, and so does not need a weighted mean between classes, but just a sample mean.

2.3.2. Dice Coefficient

Sørensen-Dice coefficient (DICE) is one of the most used metrics in clinical segmentation works and belongs to the class of overlap based metrics. From a graphic point of view, DICE considers all the elements correctly predicted (small blue square) and relates them to the maximum number of correct predictions (big blue squares) as can be seen in Figure 2.6.

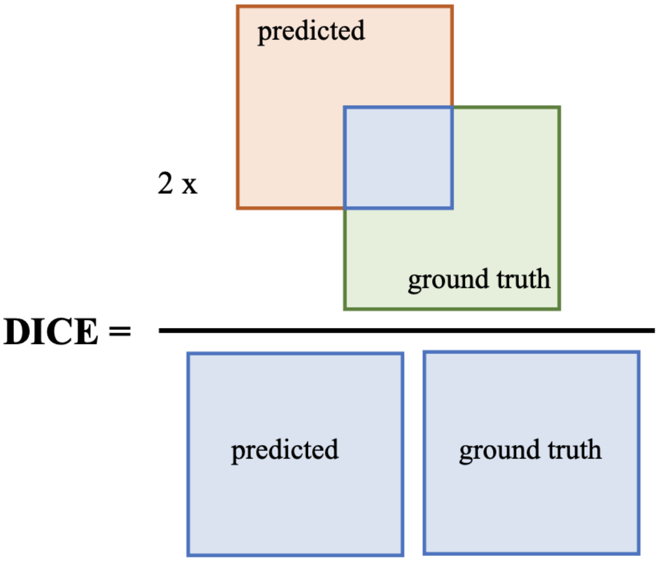


Figure 2.6: Graphics behaviour of DICE coefficient. Blue areas are those considered.

This metric can be calculated either through the four cardinalities or through direct formulas.
Regarding the first approach, considering that both the label G (ground truth) and the prediction P are crisp, the (2.5) is used to evaluate the four basic cardinalities

$$m_{ij} = \sum_{r=1}^{|X|} f_g^i(x_r) f_p^j(x_r), \tag{2.5}$$

where $|X|$ denotes the number of pixels of each segmentation, $f_g^i(x_r) = 1$ if $x_r \in G$, $f_p^i = 1$ if $x_r \in P$ and m_{ij} represent respectively TP if m_{11}, FP if m_{10}, FN if m_{01} and TN if m_{00}. These four cardinalities reflect the degree of overlap between label and prediction and produce the confusion matrix [32]. DICE evaluated in this way is given by:

$$DICE = \frac{2TP}{2TP + FP + FN}, \tag{2.6}$$

But, since in our segmentation task the labels were crisp while predictions were fuzzy, it would have been necessary to manually assign a threshold to correctly calculate the four values.
To avoid manual thresholding, the second approach, based on a direct formula, was used [35]:

$$DICE = \frac{2|G \cap P|}{|G| + |P|}, \tag{2.7}$$

From computational point of view, the implementation of DICE was divided into numerator and denominator. The numerator $2|G \cap P|$ of (2.7) was evaluated by a pixel-wise multiplication (Hadamard product) between ground truth and prediction, followed by the sum reduction of the resulting matrix. In order to quantify $|G|$ and $|P|$ the pixel values of each matrix were summed. Considering that DICE doesn't suffer from class imbalance and that seven classes were present, the overall dice was obtained as:

$$DICE = \frac{1}{C} \sum_{c=1}^{C} \frac{2 \sum_{i=1}^{N} g_i^c p_i^c}{\sum_{i=1}^{N} g_i^c + \sum_{i=1}^{N} p_i^c}, \tag{2.8}$$

where $G = \{G^1, G^2, ..., G^7\}$ are ground truth segments, $P = \{P^1, P^2, ..., P^7\}$ are prediction segments, $g_i^c \in G^c$ and $p_i^c \in P^c$ are pixel values and N is the number of pixels into the grid space. The DICE values fall into a range between [0,1], where 0 represents bad overlapping and 1 the best overlapping.
This metric is positively correlated at 52% with HD, so using both, it was possible to quantify the goodness of the predictions based on the overlap and boundaries, having a broader and more robust evaluation criterion and providing complementary information content.

2.3.3. Global Consistency Error

Global Consistency error is the last of the three metrics that were used in this study and belongs, like the DICE coefficient, to the overlap based metrics. This metric is able to focus particularly on the overlap of small portions of the images and it takes into account also the amount of true negatives. As previously said, these metrics can be expressed either directly or through the four basic cardinalities. In the direct calculation we consider the ground truth (G), the respective prediction (P) and the term $R(S, x)$, which represents the set of all the pixels belonging to S that are in the same region of x. So it is possible to define two non-symmetrical errors E at pixel x as:

$$\begin{cases} E(G,P,x) = \dfrac{|R(G,x)\backslash R(P,x)|}{|R(G,x)|}, & (2.9a) \\ E(P,G,x) = \dfrac{|R(P,x)\backslash R(G,x)|}{|R(P,x)|}, & (2.9b) \end{cases}$$

and from a graphical point of view as Figure 2.7. The GCE is hence defined as the minimum of the average of the two errors evaluated on all pixels of the image and is given by:

$$GCE(G,P) = \frac{1}{N} min \left\{ \sum_{i=1}^{N} E(G,P,x_i), \sum_{i=1}^{N} E(P,G,x_i) \right\}, \tag{2.10}$$

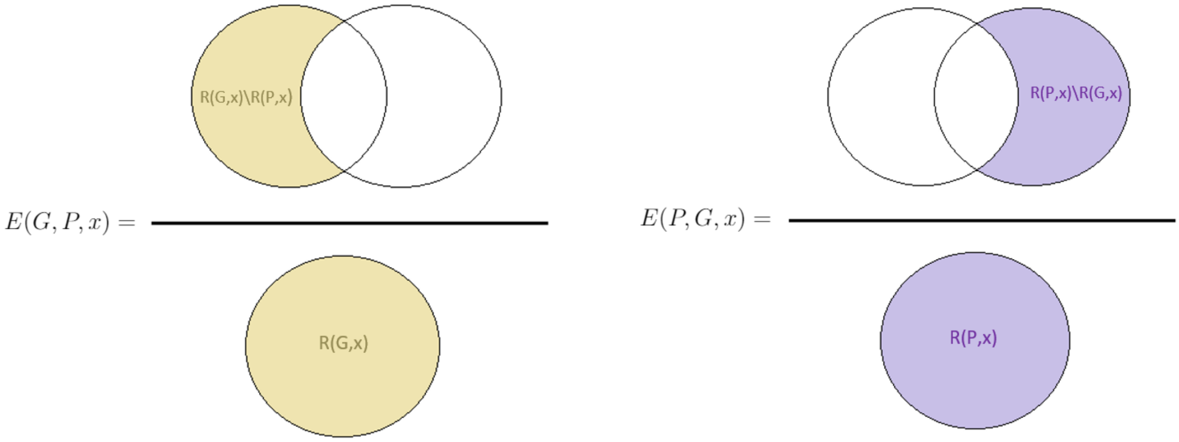


Figure 2.7: Graphical meaning of GCE components. Yellow and purple areas are those considered.

The second approach, based on cardinalities, is the one which was used in this work to reduce the computational cost of the direct one. Furthermore, considering that segmentations are made of seven classes and this method suffers from the problem of class imbalance, the overall GCE was obtained using a weighted mean among classes:

$$GCE = \frac{\sum_{c=1}^{C} w_c \frac{1}{N} min \left\{ \frac{FN(FN+2TP)}{TP+FN} + \frac{FP(FP+2TN)}{TN+FP}, \frac{FP(FP+2TP)}{TP+FP} + \frac{FN(FN+2TN)}{TN+FN} \right\}}{\sum_{c=1}^{C} w_c} \tag{2.11}$$

To avoid special cases in which the two terms of the GCE were undefined (e.g. TP+FN

$= 0$), an epsilon was added. Computational optimization given by the use of cardinalities involved the necessary selection of a threshold to bring fuzzy predictions to a crisp domain. To do this, the distribution of pixels in the various predictions was evaluated, selecting 0.8 as the optimal threshold. GCE values fall between [0,1] where 0 means the best and 1 the worst prediction. Eventually, it is not highly correlated with both the two previous metrics, keeping complementary information.

2.3.4. Overall Metric

In order to evaluate the quality of the segmentations, it was necessary to introduce an overall metric (OM) that could combine all the information provided by DICE, GCE and HD. Since the quality of predictions increases when DICE → 1 while GCE,HD → 0, the final metric was obtained as follows:

$$OM = (1 - DICE) + GCE + HD \tag{2.12}$$

This metric summarizes the aforementioned described ones and it was used for evaluating the overall goodness of networks and architectures used.

2.3.5. Precision and Recall

Two more metrics were used to evaluate the segmentations after the post-processing, together with the four just described. They are Precision and Recall defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2.13}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.14}$$

2.4. Loss Functions

Loss function is a fundamental element, on which the trend of predictions largely depends. Its primary purpose is to guide the network during training towards a configuration of parameters (weights and bias) such that the output (prediction) is as similar as possible to the ground truth. For this reason their choice is of crucial importance, since not all loss functions can consistently achieve the best performances. As said in Subsection 1.2.1, loss function surface must have a global minimum, or at least a local one, that minimizes

the differences between prediction and ground truth.
Up to now there are many specialized loss functions for different tasks, and in the case of segmentation they can be divided into three fundamental classes, based on their operating method: *Distribution-based losses*, *Region-based losses* and *Compound losses.* This classes are represented in Figure 2.8.

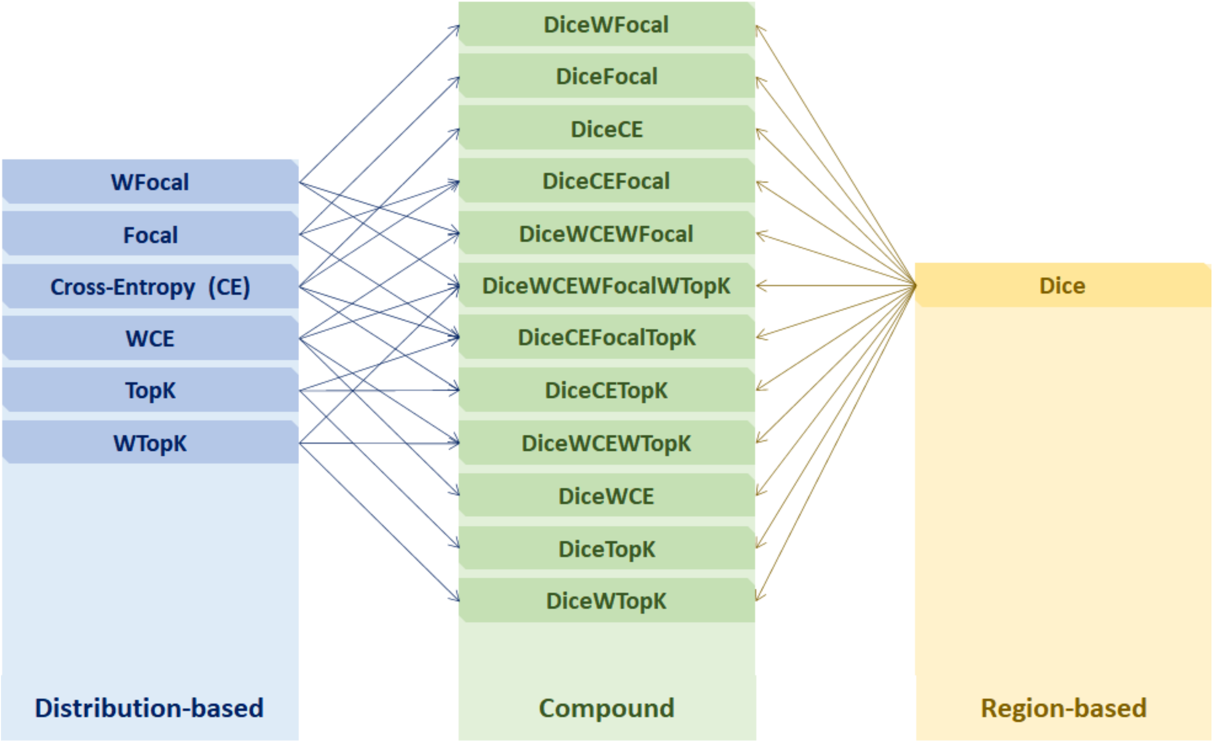


Figure 2.8: Graphical representation of the different classes of loss functions.

Specifically, $p_i^c \in P$ is the prediction pixel of class $c \in C$ and $g_i^c \in G$ the corresponding ground truth pixel, with a grid space of $N \times N$ pixels ($N = 256$) and $C = 8$ classes, where the eighth is the background class. Based on the previous consideration, in this segmentation task all the losses were developed to focus on the foreground pixels of each class, considering the background as an area belonging to other classes; for this reason the losses were developed with a categorical approach, so as to minimize the relative error related to the True Positive, and the overall loss among the classes is the sample mean between the single losses evaluated on each foreground. Here below there is the detailed description of the three classes.

2.4.1. Distribution-based losses

This class includes all the losses that aim to minimize the difference between two distributions. For multi-class segmentation these losses can be weighted if the problem of class imbalance subsists [36]. In this works the distribution-based losses that were used are: Cross-Entropy loss, TopK loss, Focal loss and their weighted versions.

Cross-Entropy loss

Cross-Entropy (CE) loss comes from the concept of relative entropy, also known as Kullback-Leibler (KL) divergence, which is a statistical distance measure of how far is the reference distribution R from the actual distribution Q as defined in 2.15.

$$\begin{aligned} D_{KL}(R||Q) &= \sum_i r_i \log \frac{r_i}{q_i} \\ &= -\sum_i r_i \log q_i + \sum_i r_i \log r_i \\ &= H(R,Q) - H(R) \end{aligned} \tag{2.15}$$

In the above formula, $H(R)$ is the entropy of distribution R while $H(R,Q)$ is the cross entropy of R and Q. To minimize the KL divergence is sufficient to minimize the H(R,Q) cross-entropy term which, in segmentation tasks, can be rewritten as in 2.16.

$$L_{CE} = -\sum_{c=1}^{C} \frac{1}{N} \sum_{i=1}^{N} g_i^c \log p_i^c \tag{2.16}$$

This form of CE is generalized and can be adopted both for crisp, fuzzy and hybrid (crisp G, fuzzy P) segmentation tasks.

Weighted Cross-Entropy loss

Weighted Cross-Entropy (WCE) loss is a weighing of CE which, at the theoretic level, should make easier the optimization of the loss in case of class imbalance, computing a weighted sum instead of sample sum between the C classes. In this specific task, a set of weights $w_c \in W$ was adopted [31], given by:

$$w_c = \frac{npix_c}{Npix}, \tag{2.17}$$

where $npix_c$ represents the number of pixels belonging to the foreground of class c, while $Npix$ is the sum of the total pixels belonging to the different classes foregrounds. This kind of weighing is used for other loss functions across the entire study.
The generalized definition of WCE is defined in 2.18.

$$L_{WCE} = -\sum_{c=1}^{C} w_c \frac{1}{N} \sum_{i=1}^{N} g_i^c \log p_i^c \tag{2.18}$$

TopK loss

TopK (TK) is a loss function deriving from Cross-Entropy which focuses no longer on the entire distribution, but on a sub-distribution K of particularly difficult pixels. These pixels can be seen as those points that the CE fails to classify under the same conditions, maintaining a significant error.
There are many methods to select the K sub-set, including unsafe methods that involve a threshold on the error between G and P. To avoid manual thresholding it was first obtained the error distribution $|P - G|$ and then selected only the pixels with values over the 95Th percentile, that was the set K of worst predictions. The overall generalized TopK is given by 2.19.

$$L_{TopK} = -\sum_{c=1}^{C} \frac{1}{N} \sum_{i \in K} g_i^c \log p_i^c \tag{2.19}$$

This loss was tested also in its weighted form, using weights defined in 2.17:

$$L_{WTopK} = -\sum_{c=1}^{C} w_c \frac{1}{N} \sum_{i \in K} g_i^c \log p_i^c \tag{2.20}$$

Focal loss

Focal loss (FL) is derived from Cross-Entropy and its main role is to focus on the worst classified samples. It can overcome the class-imbalance problem and is defined as:

$$L_{Focal} = -\sum_{c=1}^{C} \frac{1}{N} \sum_{i=1}^{N} (1 - p_i^c)^{\gamma} g_i^c \log p_i^c, \tag{2.21}$$

were $\gamma = 2$ obtained the best performances, according to [36].

This loss was tested also in its weighted form, using weights defined in 2.17:

$$L_{WFocal} = -\sum_{c=1}^{C} w_c \frac{1}{N} \sum_{i=1}^{N} (1 - p_i^c)^\gamma g_i^c \log p_i^c, \tag{2.22}$$

2.4.2. Region-based losses

Losses included in this class measure the difference between ground truth G and prediction P trying to maximize the overlap or to minimize the mismatch between the two. In this work the region-based loss that was used is Dice loss.

Dice loss

Dice loss (DL) derives from the Sørensen-Dice coefficient (DICE), defined in Subsection 2.3.2, which is modified in order to be minimized [35]. To be used as a loss function, DICE must tend to zero for optimal cases and for this reason it is reversed as follows:

$$DL = 1 - DICE, \tag{2.23}$$

that, in multi-class problem is given by:

$$L_{Dice} = \frac{1}{C} \sum_{c=1}^{C} \left[1 - \sum_{i=1}^{N} \frac{2 g_i^c p_i^c}{g_i^c + p_i^c} \right] \tag{2.24}$$

Moreover, the DL cannot be implemented in a weighted version, such as distribution-based losses, because its evaluation range falls between [0,1] whatever the foreground/background imbalance, as it evaluates only the degree of foreground overlapping.

2.4.3. Compound losses

Compound losses are conceived as the sum of multiple losses, with the aim of improving performance by combining the different strengths. In the literature there are compound losses composed of a maximum of two elements [36], while in this study this concept was extended to three and four elements.

The two-elements losses that were used and then tested were the combination between Dice loss (first term) and each loss belonging to the distribution-based class (second term). The multi-elements, on the other hand, combine more than two losses.

Dice-CrossEntropy

Dice-CrossEntropy loss (DCE) and its weighted version are given by 2.25a, 2.25b respectively.

$$L_{DiceCE} = L_{Dice} + L_{CE}, \tag{2.25a}$$

$$L_{DiceWCE} = L_{Dice} + L_{WCE} \tag{2.25b}$$

From an operational point of view this loss takes into account the dissimilarities between the two distributions and also the overlap degree.

Dice-TopK

Similar to Dice-CrossEntropy, Dice-TopK (DTopK) is the combination between Dice loss and TopK loss. Unlike DCE, it evaluates only the distribution related to the worst subset of the prediction errors and was used and tested in its original 2.26a and weighted version 2.26b.

$$L_{DiceTopK} = L_{Dice} + L_{TopK}, \tag{2.26a}$$

$$L_{DiceWTopK} = L_{Dice} + L_{WTopK} \tag{2.26b}$$

Dice-Focal

This loss function allows the network to learn from the worst champions and considerably alleviates class imbalance thanks to the combined action of the two losses that compose it (both do not suffer from class imbalance). Dice-Focal (DF) was implemented and tested in its original 2.27a and weighted version 2.27b.

$$L_{DiceFocal} = L_{Dice} + L_{Focal}, \tag{2.27a}$$

$$L_{DiceWFocal} = L_{Dice} + L_{WFocal} \tag{2.27b}$$

Dice-CrossEntropy-TopK

Dice-CrossEntropy-TopK is a multi-compound loss and integrates the best properties of DL, CE and TopK. It was evaluated in both original 2.28a and weighted version 2.28b.

$$L_{DiceCETopK} = L_{Dice} + L_{CE} + L_{TopK}, \tag{2.28a}$$

$$L_{DiceWCETopK} = L_{Dice} + L_{WCE} + L_{WTopK} \tag{2.28b}$$

Dice-CrossEntropy-Focal

Dice-CrossEntropy-Focal is a multi-compound loss and integrates the best properties of DL, CE and Focal. It was evaluated in both original 2.29a and weighted version 2.29b.

$$L_{DiceCEFocal} = L_{Dice} + L_{CE} + L_{Focal}, \tag{2.29a}$$

$$L_{DiceWCEFocal} = L_{Dice} + L_{WCE} + L_{WFocal} \tag{2.29b}$$

Dice-CrossEntropy-Focal-TopK

This is the last compound loss that was tested and it combines the properties of all the one-element losses. This leads to a function that takes into account all the major aspects of the elements that compose it: complete check of distribution and overlap (CE, DL), focus on the hard samples (TopK, Focal), alleviation of class imbalance (Focal, DL), attention to the highest errors (TopK).
The loss was implemented in both original 2.30a and weighted version 2.30b.

$$L_{DiceCEFocalTopK} = L_{Dice} + L_{CE} + L_{Focal} + L_{TopK}, \tag{2.30a}$$

$$L_{DiceWCEFocalTopK} = L_{Dice} + L_{WCE} + L_{WFocal} + L_{WTopK} \tag{2.30b}$$

2.4.4. Other losses

One particular loss function that was used to train a branch of the CEL-UNet architecture, that will be described in Subsection 2.5.5, is the Edge loss (EL) but its use is relegated to the sole function of training the network, therefore it was not subject to performance tests.

Edge loss

The Edge loss is a weighted combination of two particular cross-entropies, as defined in 2.31.

$$L_E = [\beta C + (1 - \beta)\hat{C}] \tag{2.31}$$

In the above function C, $\hat{C}$ and β are defined as follows:

$$C = -\sum_{c=1}^{C}\sum_{i=1}^{N} DWM_c c_i^c \log p_i^c, \tag{2.32a}$$

$$\hat{C} = -\sum_{c=1}^{C}\sum_{i=1}^{N} DWM_c (1 - c_i^c) \log (1 - p_i^c), \tag{2.32b}$$

$$\beta = 1 - \frac{Ncontpix}{Npix} \tag{2.32c}$$

C and $\hat{C}$ are called cross-entropy and reverse cross-entropy and they use respectively the contours c_i^c and the negative contours $(1-c_i^c)$ of the true labels G_c as new ground truth to pilot the net to predict contours instead of areas. $Ncontpix$ is the total number of pixels belonging to the contours, while $Npix$ is the overall number of pixels in the batch. Moreover, DWM is a negative exponential transformation of the euclidean distance transform (EDT), which assigns to each pixel of the grid space its distance value from the label boundary. These elements are depicted in Figure 2.9. DWM_c is defined as:

$$DWM_c = \left[1 + \gamma \exp\left(-\frac{EDT_c}{\sigma}\right)\right] \tag{2.33}$$

where $\gamma = 8$ and $\sigma = 10$ were selected as better parameters. DWM was used to enhance the contours and the near-surface pixels, obtaining a focus on this particular regions.

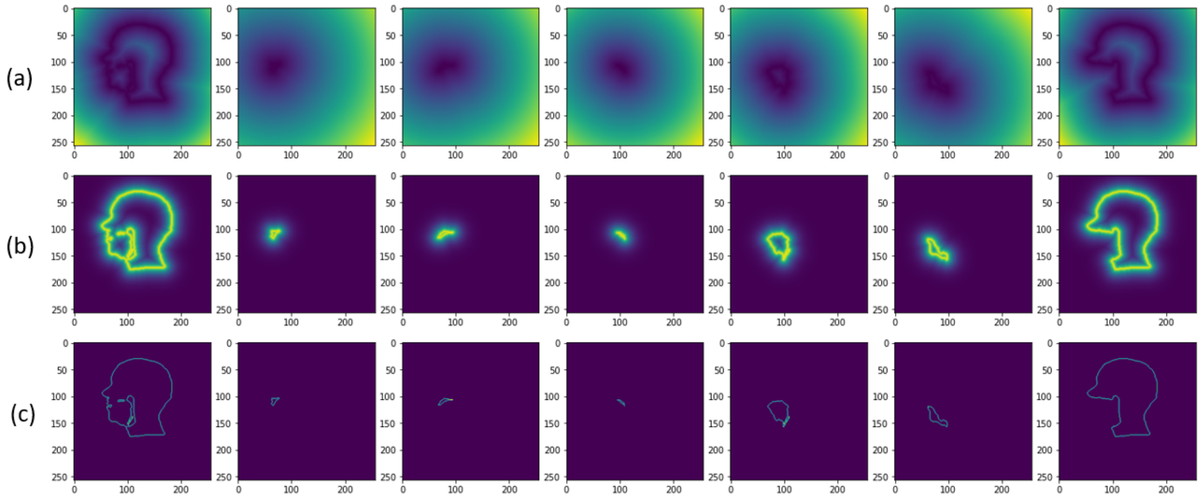


Figure 2.9: Components of Edge Loss: (a) EDT, (b) DWM and (c) represents the contours c (its negative is $(1 - c)$).

2.4.5. Loss functions behaviour comparison

In Figure 2.10 are depicted the trends of the aforementioned loss functions taking into account one single pixel p_i^c of the image with ground truth value g_i^c equal to 1 (pixel belonging to the foreground). The graphic shows the trend of the losses in function of the predicted value of the considered pixel p_i^c. It must be noted that: only non-weighted loss functions are represented because weights are computed just to deal with class imbalance, so they don't affect the prediction of the single pixel; Cross Entropy loss and TopK loss are identical considering the single pixel.

It can be noted that single loss functions don't have high values in correspondence of the worst classification case (when $p_i^c = 0$) and also their slope is quite low. Their value decreases rapidly and the curve becomes practically horizontal when approaching the best classification case (when $p_i^c = 1$). On the other hand, the more the loss functions are combined the higher is the initial value and the slope, which is also kept until the best classification is reached. From this evidence it is possible to suppose that compound loss functions learn better than single ones during the whole network learning.

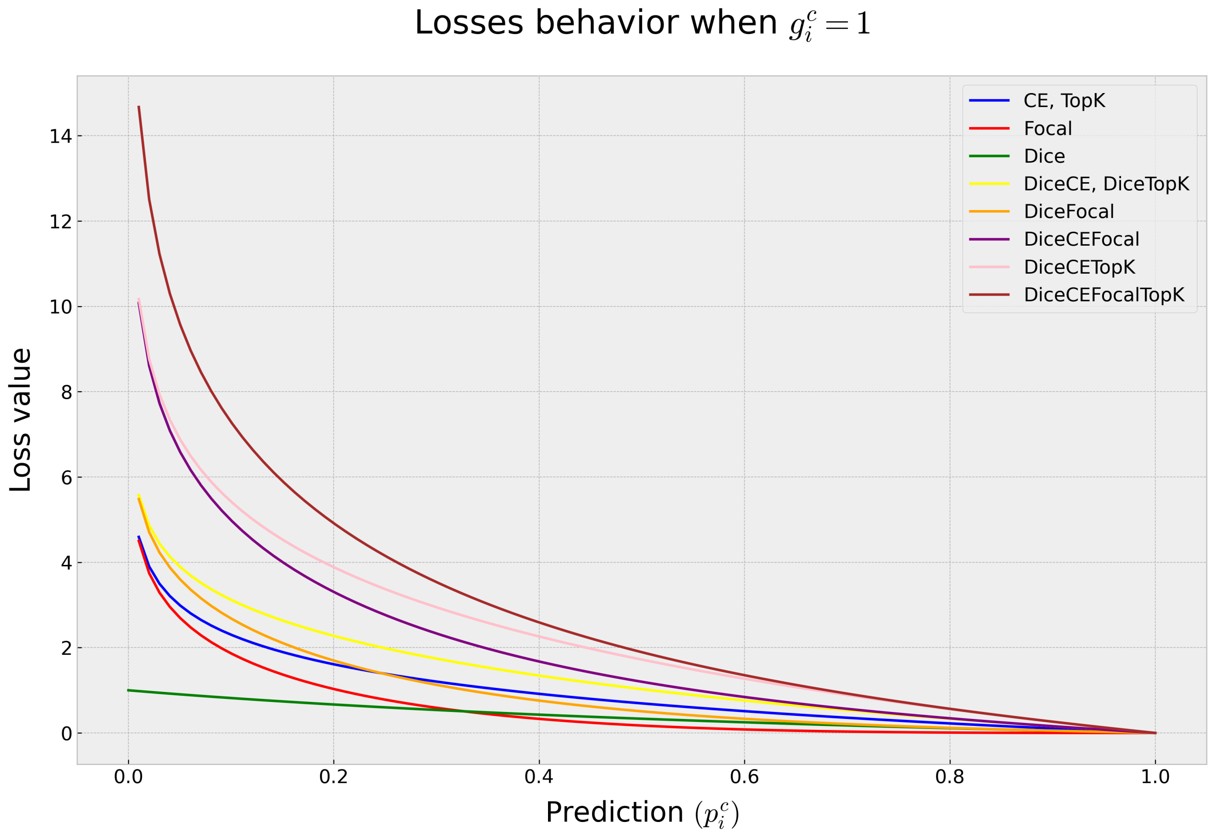


Figure 2.10: Loss functions behaviour for a single pixel belonging to the foreground.

2.5. Architectures

Five different U-shaped Convolutional Neural Networks with different configurations were developed to accomplish segmentation task. All Networks received batches of MRI images as input I and produced the seven corresponding labelled images as output O. During the training phase the batches were extracted from the train dataset, while during the validation phase the batches were taken from the validation set.
The five architectures used are described below, and divided into *encoding path*, *bottleneck* and *decoding path*.

2.5.1. Ruthven-UNet

Ruthven-UNet is a UNet developed for vocal tract segmentation task and presented in [31].
The ***encoding path*** is composed of a sub-sequence of $e = 4$ encoding processing blocks (ePB) which give the network a depth $dp = 4$ and analyze the input I in a receptive field that becomes gradually larger as a function of the level e. In particular, each ePB is composed of two sub-blocks in series that include:

- one convolution layer with kernel size $K_c = (3 \times 3)$, stride and padding calibrated to maintain the same grid-space, activation function *linear* and F_e filters;
- one batch normalization layer;
- one ReLU activation layer.

The batch normalization and ReLU combination helps the net to prevent overfitting easing the optimization.
The two sub-blocks are followed by a max-pooling layer with kernel size $K_{mp} = (2 \times 2)$ and stride $S_{mp} = (2, 2)$, which reduce the input grid-size. Each ePB, before the max-pooling layers, extracts its output to be projected in the parallel decoding branch with a skip connection. The filter size F_e of each convolution starts from $F_1 = 64$ and increases level by level with a function given by $F_e = F_1 \cdot 2^e$; in this setting the output of the *encoding path* is a series of $F_4 = 512$ feature maps with grid-size of 16×16.
In this architecture ***bottleneck*** is the deepest part of the network that contains the maximum level of encoding and is made up of a block equal to the ePB, but in this case, the max-pooling down-sampling layer is not used. The *bottleneck* output is a series of $F_5 = 1024$ feature maps with grid-size of 16x16.
The ***decoding path*** task is to synthesize the knowledge extracted from the *encoding path* to produce the correct output O. This path is symmetrical to the *encoding path* and

composed of $d = 4$ decoding processing blocks (dPB). In particular, each dPB is designed as follows:

- one transpose convolution (deconvolution) layer which counterbalances the parallel max-pooling layer down-sampling with grid-size up-sampling. It is composed of kernel size $K_d = (2 \times 2)$, stride $S_d = (2, 2)$, activation *linear* and F_d filters;
- one concatenation layer that concatenates together the ePB output from the parallel skip connection and the deconvolution output;
- one ePB with F_d filters and without max-pooling, to analyze the concatenation information.

The filter size F_d of each dPB starts from $F_1 = 512$ and decreases level by level with a function given by $F_d = F_1 \cdot 2^{(dp-1)d}$. When it reaches depth $d = 4$ the output block occurs and, in this particular case, one convolutional layer with *soft-max* activation is added to the last dPB to produce the seven output probability maps.
Ruthven U-Net architecture is depicted in Figure 2.11.

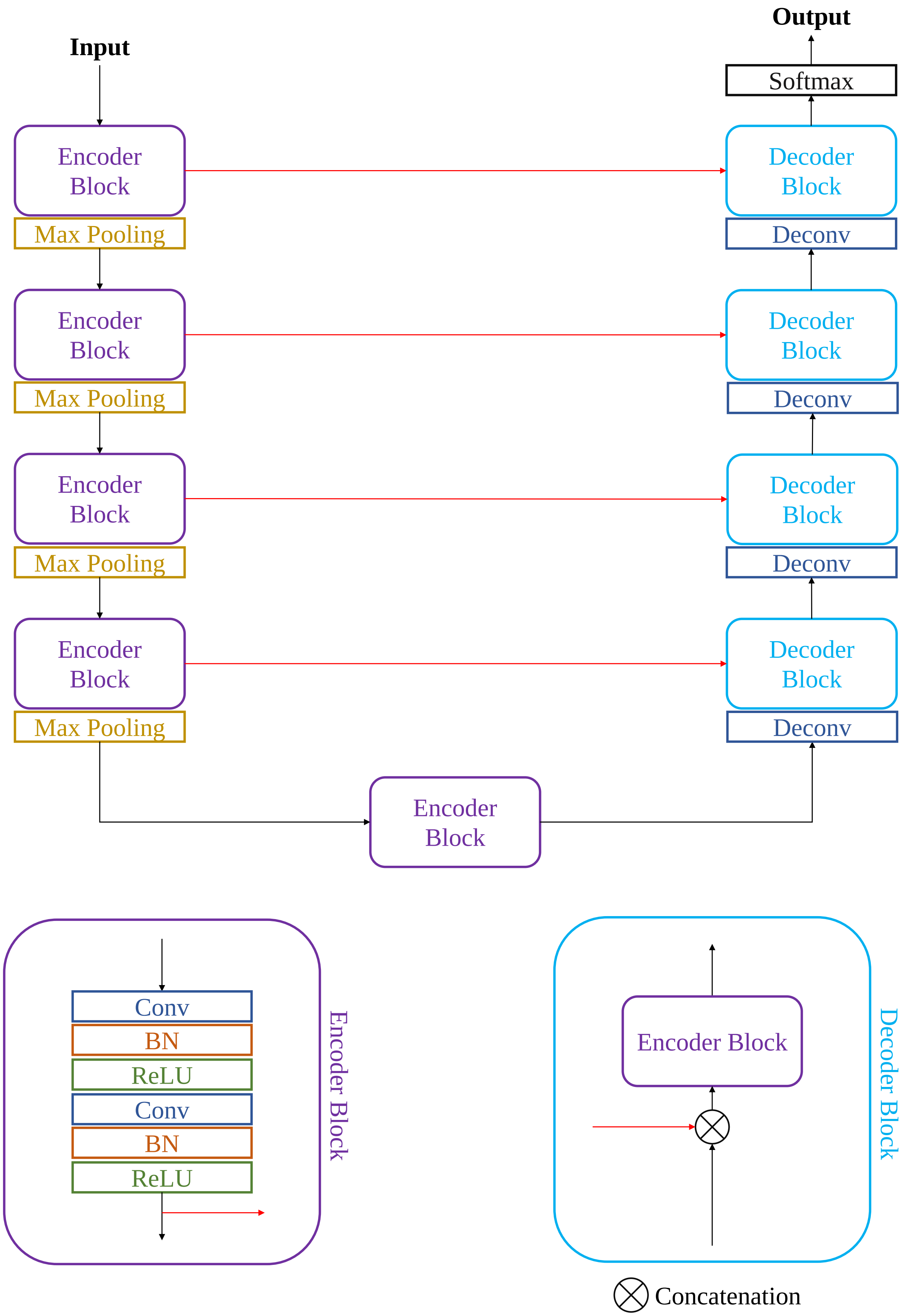


Figure 2.11: Graphical representation of Ruthven-UNet architecture."Conv" and "Deconv" blocks are intended with kernel size 3 × 3.

2.5.2. QT-UNet

Quick Tumor UNet is a U-shaped CNN developed by [37] to localize and segment three different types of tumors in the brain (meningioma, glioma and pituitary brain tumors). In this work, QT-Net was adapted to correctly work with the current input-output set-up and to accomplish the vocal-tract segmentation task.

QT-Net has a depth $dp = 4$ and the ***encoding path*** is composed by four particular encoding processing blocks called dense blocks (eDNSB). They are composed by a series of convolutional layers:

- the 1^{st} convolutional layer with kernel size $K_e = 5 \times 5$, stride and padding calibrated to maintain the same grid-space, activation function $ReLU$ and F_e filters;
- the 1^{st} batch normalization layer. These layers stabilize the learning process and reduce the number of epochs to convergence, making mean values close to 0 and standard deviations close to 1;
- the 1^{st} concatenation layer between the 1^{st} convolutional layer input and the 1^{st} batch normalization output;
- the 2^{nd} convolutional layer with kernel size $K_e = 5 \times 5$, stride and padding calibrated to maintain the same grid-space, activation function $ReLU$ and F_e filters;
- the 2^{nd} batch normalization layer;
- the 2^{nd} concatenation layer between the 1^{st} convolutional layer input, the 2^{nd} convolutional layer input and the 2^{nd} batch normalization output;
- the 3^{rd} convolutional layer with kernel size $K_e = 1 \times 1$, stride and padding calibrated to maintain the same grid-space, activation function $ReLU$ and F_e filters;
- the 3^{rd} batch normalization layer.

The dense blocks outputs are sent, by mean of skip-connections, to the parallel levels in the *decoding path*. Moreover, in the analysis path, the blocks are followed by a max-pooling layer with kernel size $K_{mp} = 2 \times 2$. Max-pooling operates a down-sampling of the image to prevent the network from a large number of learnable parameters (weights and biases) and so from degradation. The filter size F_e is kept constant for each convolutional layer of each eDNSB and set to $F_e = 64$; in this particular configuration, the filter size is also called growth-rate of the net and can be used to easily check the parameter space of the network. The eDNSB configuration provides some advantages to the network such as the increase of gradient back-propagation efficiency (dense block interconnections and skip-connections). Another advantage is that the *encoding path* provides to the synthesis path (*decoding path*) one set of feature maps that contain all the preceding dense blocks inputs.

The ***bottleneck*** is conceived with one convolutional layer, with kernel size $K_e = 1 \times 1$, stride and padding calibrated to maintain the same grid-space, activation function $ReLU$ and $F_b = 64$ filters, followed by one batch normalization layer. This level is designed to reduce feature maps dimensionality maintaining the same information content, in order to facilitate the synthesis path.

The ***decoding path*** is assembled to perform the summarization of information:

- one un-pooling layer with kernel size $K_us = 2 \times 2$. This layer counterbalances the max-pooling approximation with the opposite operation and increases the grid-size similarly to deconvolution. In this case, it up-samples the input grid-size using nearest-neighbour interpolation instead of transpose convolution;
- one concatenation layer that concatenates together the eDNSB output from the parallel skip connection and the up-sampling output;
- one concatenation layer that concatenates together the eDNSB output from the parallel skip connection, and the un-pooling output;
- one eDNSB, as described above, to analyze the up-sampling operation and extract knowledge level by level. Also in this blocks the filter size of each convolutional layer is fixed to $F_d = 64$ for all levels.

When the last level occurs, the output is subjected to a 7-layers soft-max to provide the seven classes probability maps as output.

QT-UNet architecture is depicted in Figure 2.12.

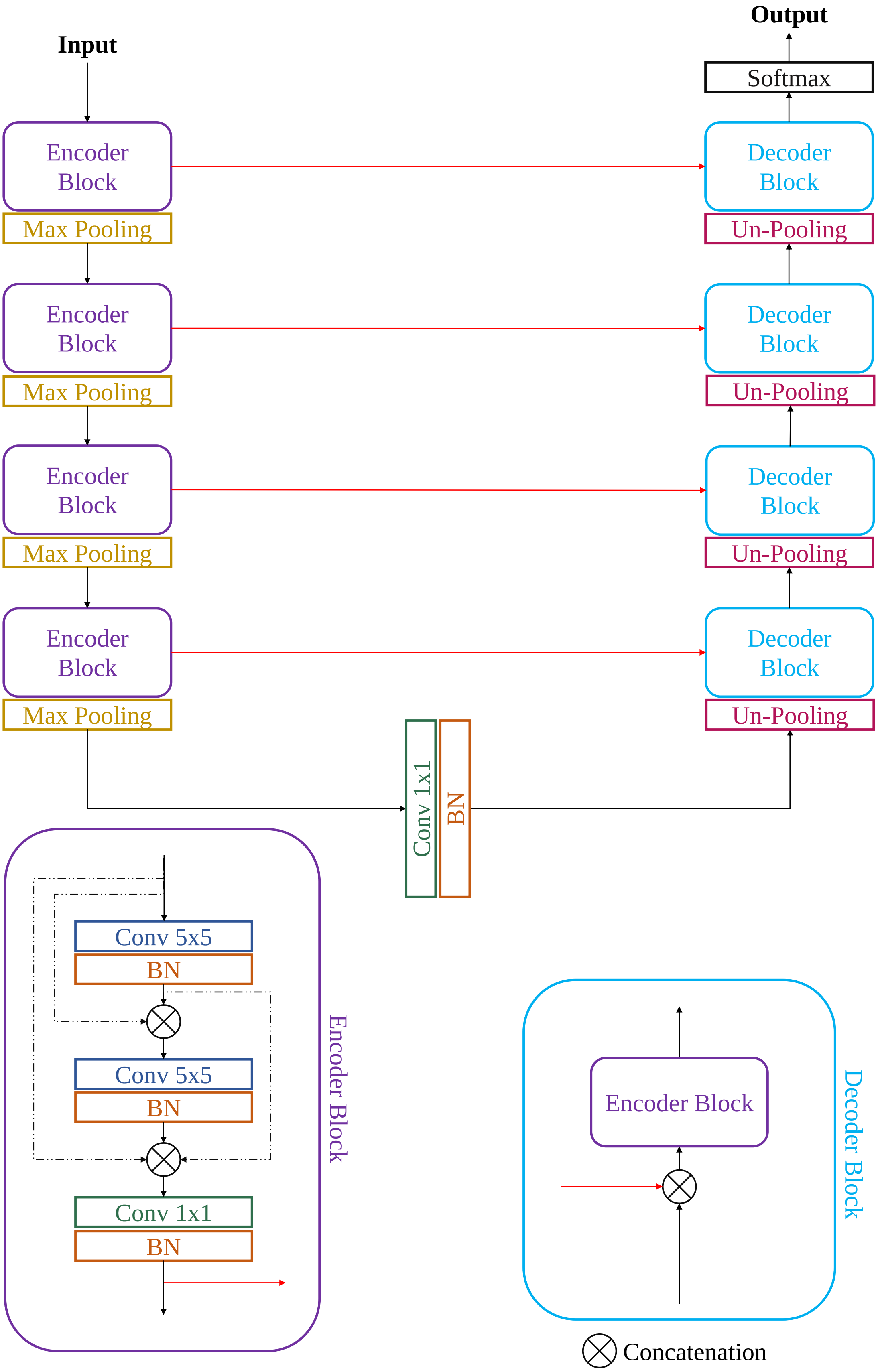


Figure 2.12: Graphical representation of QT-UNet architecture.

2.5.3. IM-UNet

IM-UNet (improved UNet) is a U-shaped CNN developed for the segmentation of human metaphase II (MII) oocyte images. This architecture was developed to overcome the problem of the limited receptive field of the classic UNet, which does not allow to grasp both the details and the global aspects of an image; this because UNet depth must be limited to avoid degradation [38]. For the aim of the thesis, the architecture was adapted to the input-output set-up and ablation was performed to find its optimal hyper-parameters.

The ***encoding path*** starts with a residual processing block (eRPB) that provides a particular input to the subsequent analysis levels. This input is not the singular 256×256 image, but a set of feature maps that propagate a multi-scale information. This first eRPB is divided in two branches and developed as follows:

- the 1^{st} convolutional layer in the 1^{st} branch with kernel size $K_e = 3 \times 3$, stride and padding calibrated to maintain the same grid-space, activation function *linear* and $F_{1r} = 8$ filters. This convolution extracts a more detailed spatial information, but less overall knowledge;
- the batch normalization layer in the 1^{st} branch;
- the ReLU activation layer in the 1^{st} branch;
- one convolutional layer in the 2^{nd} branch with kernel size $K_e = 1 \times 1$, stride and padding calibrated to maintain the same grid-space, activation function *linear* and $F_{2r} = 8$ filters. This convolution provides a global knowledge of the network given that 1×1 convolution is a linear projection of the input into a stack of feature maps that maintain the overall informative content;
- one batch normalization layer in the 2^{nd} branch.

The residual block effectiveness is verified only if the number of feature maps of each convolutional layer is kept low and the outputs of the two branches are added pixel-wise to be provided to the next levels.

After the eRPB, there is a series of 4 encoding processing blocks (ePB) (IM-UNet depth is $dp = 4$) that are designed as follows:

- the 1^{st} batch normalization layer in the 1^{st} branch;
- the 1^{st} ReLU activation layer in the 1^{st} branch;
- the 1^{st} convolutional layer in the 1^{st} branch with kernel size $K_e = 3 \times 3$, stride and padding calibrated to maintain the same grid-space, activation function *linear* and F_e filters;

- the 2^{nd} batch normalization layer in the 1^{st} branch;
- the 2^{nd} ReLU activation layer in the 1^{st} branch;
- the 2^{nd} convolutional layer in the 1^{st} branch with kernel size $K_e = 3 \times 3$, stride and padding calibrated to maintain the same grid-space, activation function *linear* and F_e filters;
- one convolutional layer in the 2^{nd} branch with kernel size $K_e = 1 \times 1$, stride and padding calibrated to maintain the same grid-space, activation function *linear* and F_e filters;
- one batch normalization layer in the 2^{nd} branch;
- one pixel-wise summation layer that combines the knowledge from the two previous branches;
- one skip-connection that sends the information to the parallel synthesis level;
- one max-pooling layer with kernel size $K_{mp} = 2 \times 2$ and stride $S_{mp} = (2,2)$, to prevent degradation and lower the network complexity. It also down-samples the grid size of the feature maps;
- one dropout layer with a dropout rate $dr = 0.5$ to lower the generalization error and prevent overfitting, disabling 50% of its weights at each epoch.

The filter size F_e of each convolution starts from $F_1 = 64$ and increases level by level with a function given by $F_e = F_1 \cdot 2^e$; in this setting the output of the *encoding path* is a series of $F_4 = 512$ feature maps with grid-size of 16×16.

The ***bottleneck***, also called bridge, is conceived with a particular set of convolutional layers called dilated convolutions. This architecture section is very valuable because it allows to further increase the receptive field without further max-pooling, in order to avoid the reduction of resolution. In this work four dilated convolutions are placed in series, producing at the output some denser feature maps than the starting ones, which provide more context to the *decoding path*, while maintaining a high level of detail. The four dilated convolutions bridge is implemented as follows:

- the 1^{st} dilated convolutional layer with kernel size $K_b = 1 \times 1$, stride and padding calibrated to maintain the same grid-space, activation function *linear*, $F_b = 1024$ filters and dilatation rate $dilr = 2$;
- the 2^{nd} dilated convolutional layer with kernel size $K_b = 1 \times 1$, stride and padding calibrated to maintain the same grid-space, activation function *linear*, $F_b = 1024$

filters and dilatation rate $dilr = 4$;

- the 3^{rd} dilated convolutional layer with kernel size $K_b = 1 \times 1$, stride and padding calibrated to maintain the same grid-space, activation function *linear*, $F_b = 1024$ filters and dilatation rate $dilr = 6$;
- the 4^{th} dilated convolutional layer with kernel size $K_b = 1 \times 1$, stride and padding calibrated to maintain the same grid-space, activation function *linear* ,$F_b = 1024$ filters and dilatation rate $dilr = 8$.

The ***decoding path*** is developed in a symmetric and complementary way with respect to the *encoding path*. It synthesizes the knowledge into the seven classes with a series of 4 decoding processing blocks (dPB) described below:

- one transpose convolution (deconvolution) layer that counterbalances the parallel max-pooling layer down-sampling with a grid-size up-sampling. It is composed of kernel size $K_d = (1 \times 1)$, stride $S_d = (2, 2)$, activation *linear* and F_d filters;
- one batch normalization layer;
- one concatenation layer that puts together the transpose convolution output and the feature maps from the parallel skip connection;
- one dropout layer with a dropout rate $dr = 0.5$;
- the 1^{st} batch normalization layer in the 1^{st} branch;
- the 1^{st} ReLU activation layer in the 1^{st} branch;
- the 1^{st} transpose convolutional layer in the 1^{st} branch with kernel size $K_e = 3 \times 3$, stride and padding calibrated to maintain the same grid-space, activation function *linear* and F_d filters;
- the 2^{nd} batch normalization layer in the 1^{st} branch;
- the 2^{nd} ReLU activation layer in the 1^{st} branch;
- the 2^{nd} transpose convolutional layer in the 1^{st} branch with kernel size $K_e = 3 \times 3$, stride and padding calibrated to maintain the same grid-space, activation function *linear* and F_d filters;
- one transpose convolutional layer in the 2^{nd} branch with kernel size $K_e = 1 \times 1$, stride and padding calibrated to maintain the same grid-space, activation function *linear* and F_d filters;
- one batch normalization layer in the 2^{nd} branch;

- one pixel-wise summation layer that combines the synthesized knowledge from the two previous branches.

The filter size F_d of each dPB starts from $F_1 = 512$ and decreases level by level with a function given by $F_d = F_1 \cdot 2^{(dp-1)d}$. When the last level occurs, the output is subjected to a 7-layers soft-max to provide the seven classes probability maps as output.
IM-UNet architecture is depicted in Figure 2.13.

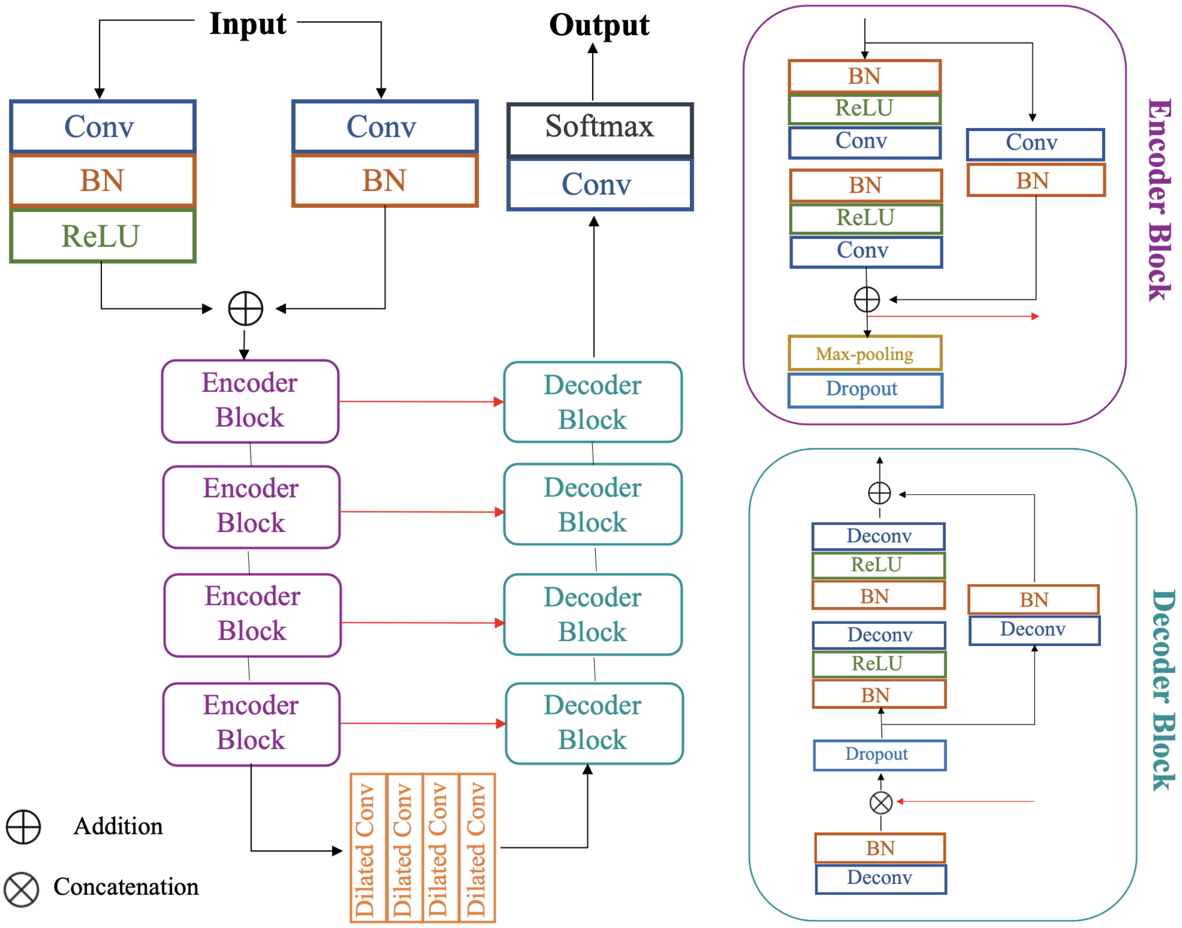


Figure 2.13: Graphical representation of IM-UNet architecture.

2.5.4. IM-UNet with attention block

The concept of "attention", as explained by [39], regards the capability of the network to focus only to certain portions of the image, gaining a higher generalizability. The type of attention implemented inside the traditional IM-UNet is called "soft attention" and it implies that larger weights are attributed to more relevant portions of the image, while smaller weights are attributed to less relevant regions. This way, during training, the network focuses more on highly weighted areas. Attention block is introduced in correspondence of the skip connections to diminish the number of redundant features that are brought from the down-sampling path to the up-sampling path.

The attention block takes two inputs which are respectively the original skip connection (x) and the output of the last dilated convolution or the previous decoder block (g). It produces one output that is the new skip connection (*x'*).

It is composed by the following layers:

- one strided convolution for input x and one 1×1 convolution for input g;
- Element-wise summation in order to enlarge aligned weights and reduce unaligned weights;
- one ReLU activation function;
- one 1×1 convolution;
- one Sigmoid layer that produces the attention coefficients;
- one interpolation layer used to up-sample the weights to the original dimension of x;
- Element-wise multiplication between the weights and x in order to obtain the output *x'* which is scaled according to relevance.

The attention block just described together with the original IM-UNet architecture is depicted in Figure 2.14.

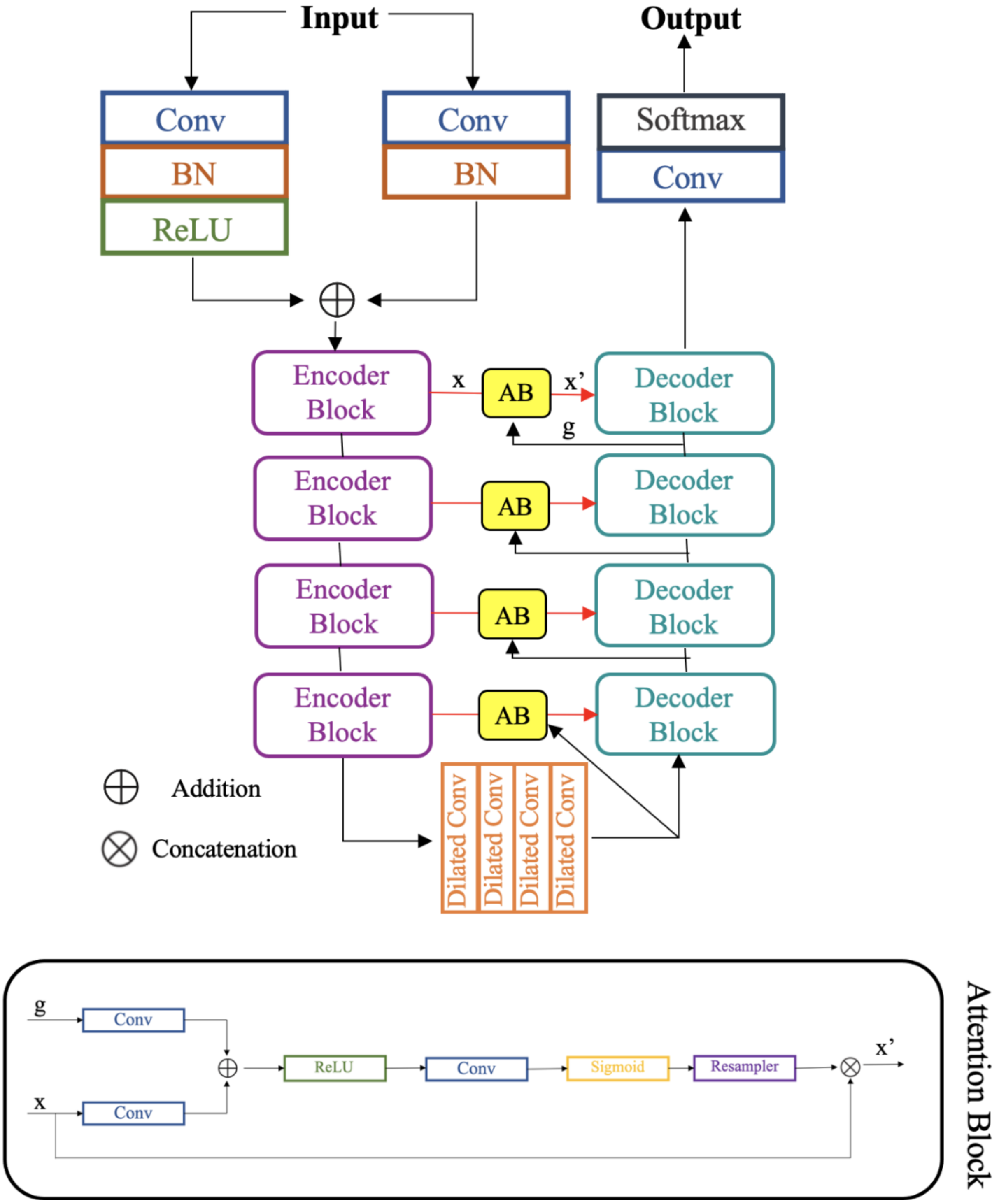


Figure 2.14: Graphical representation of IM-UNet attention architecture.

2.5.5. CEL-UNet

CEL-UNet is an Y-shaped convolutional neural network developed for CT knee volumes segmentation by [40]. It encompasses one *encoding path* and two *decoding paths* that synthesize respectively the segmentation volume and the segmentation borders. For the purpose of the thesis, this network was readjusted to work with images, instead of volumes, and produce the correct output probability maps.

The ***encoding path*** has a depth $dp = 5$ that leads to a series of $e = 5$ encoding processing blocks (ePB), each consisting of a series of two sub-blocks, individually defined as follows:

- one convolutional layer with kernel size $K_c = (3 \times 3)$, stride $S_c = (2, 2)$ and padding $P_c = 2$ to maintain the same grid-space, activation function *linear* and F_e filters;
- one batch normalization layer;
- one ReLU activation layer.

The two sub-blocks are followed by a max-pooling layer with kernel size $K_{mp} = (2 \times 2)$ and stride $S_{mp} = (2, 2)$, which reduces the input grid-size. Each ePB, before the max-pooling layers, extracts its output to be projected in the parallel decoding branches with the relative skip connections. The filter size F_e of each convolution starts from $F_1 = 8$ and increases level by level with a function given by $F_e = F_1 \cdot 2^e$; in this setting the output of the *encoding path* is a series of $F_5 = 128$ feature maps with grid-size of 8×8.

The ***bottleneck*** contains the highest content of information and here max-pooling down-sampling layer is not used. Its output is a series of $F_6 = 256$ feature maps with grid-size of 8×8.

The output of *bottleneck* is passed to the two branches to be synthesized in two different manners to predict segmentation areas and segmentation edges. The ***edge decoding path*** is based on decoding processing block (dPB) which is designed with the architecture described below:

- one transpose convolutional (deconvolution) layer which counterbalances the parallel max-pooling layer down-sampling with grid-size up-sampling. It is composed of kernel size $K_d = (2 \times 2)$, stride $S_d = (2, 2)$, activation *linear* and F_d filters;
- one ReLU activation layer;
- one concatenation layer that concatenates together the ePB output from the parallel skip connection and the deconvolution output;
- one ePB with F_d filters and without max-pooling, to analyze the concatenation information;
- one pyramidal edge extraction block (PEE).

PEE block is based on the pyramidal edge extraction paradigm where, at first, the input I_{PEE} is linearly projected by 1×1 convolution and then the edges are evaluated at different scales. The different scales border are extracted by mean of subtraction between the convolution output and the average pooling of I_{PEE} at different scales *scl*. In this work scales are set to be $scl \in 1, 2$ and the PEE block is developed as follows:

- one convolutional layer with kernel size $K_pee = (1 \times 1)$, stride and padding calibrated to maintain the same grid-space, activation function *linear* and $F_{pee} = \frac{F_d}{2}$ filters with output O_{cPEE};
- one average-pooling layer that represents $scl = 1$ with kernel size $K_{av1} = (5 \times 5)$, stride $S_{av1} = (1, 1)$ and padding calibrated to maintain same grid size. Its output is O_{av1PEE};
- one pixel-wise subtraction layer to extract the edges at $scl = 1$ between O_cPEE and O_{av1PEE} with output E_{sc1};
- one average-pooling layer that represents $scl = 2$ with kernel size $K_{av2} = (7 \times 7)$, stride $S_{av1} = (1, 1)$ and padding calibrated to maintain same grid size. Its output is O_{av2PEE};
- one pixel-wise subtraction layer to extract the edges at $scl = 1$ between O_{cPEE} and O_{av2PEE} with output E_{sc2};
- one concatenation layer that concatenates together O_{cPEE}, E_{sc1} and E_{sc2};
- one convolutional layer with kernel size $K_pee = (1 \times 1)$, stride and padding calibrated to maintain the same grid-space, activation function *linear* and $F_{pee} = F_d$;
- one skip connection that sends the convoluted output to the post transpose convolution layer of the parallel mask branch.

The dPB is repeated for $dp = 5$ times and then subjected to a 7-layer soft-max to provide the edges of the seven classes as output which are used to train this branch by mean of edge loss defined in Subsection 2.4.4.
The parallel ***mask decoding path*** is based on decoding processing block (dMPB) which is designed with the following layers:

- one transpose convolutional (deconvolution) layer which counterbalances the parallel max-pooling layer down-sampling with grid-size up-sampling. It is composed of kernel size $K_d = (2 \times 2)$, stride $S_d = (2, 2)$, activation *linear* and F_d filters;
- one ReLU activation layer;
- one concatenation layer that concatenates together the ePB output from the parallel skip connection and the deconvolution output;

- one ePB with F_d filters and without max-pooling, to analyze the concatenation information;
- one concatenation layer that concatenates together the parallel branch PEE block output and the deconvolution output.

It must be noted that the vertical skip-connections are unidirectional, from edge to mask branch, to increase spatial scaling and merge together the knowledge extracted from the *mask decoding path* with the knowledge extracted from the *edge decoding path.* The dMPB is repeated for $dp = 5$ times and then subjected to a 7-layer soft-max to provide the areas of the seven classes as output.

As a common aspect, the two decoding branches have filter size F_d that starts from $F_1 = 256$ and decreases level by level with a function given by $F_d = F_1 \cdot 2^{(dp-1)d}$, where d is the depth level of the network.

CEL-UNet architecture is depicted in Figure 2.15.

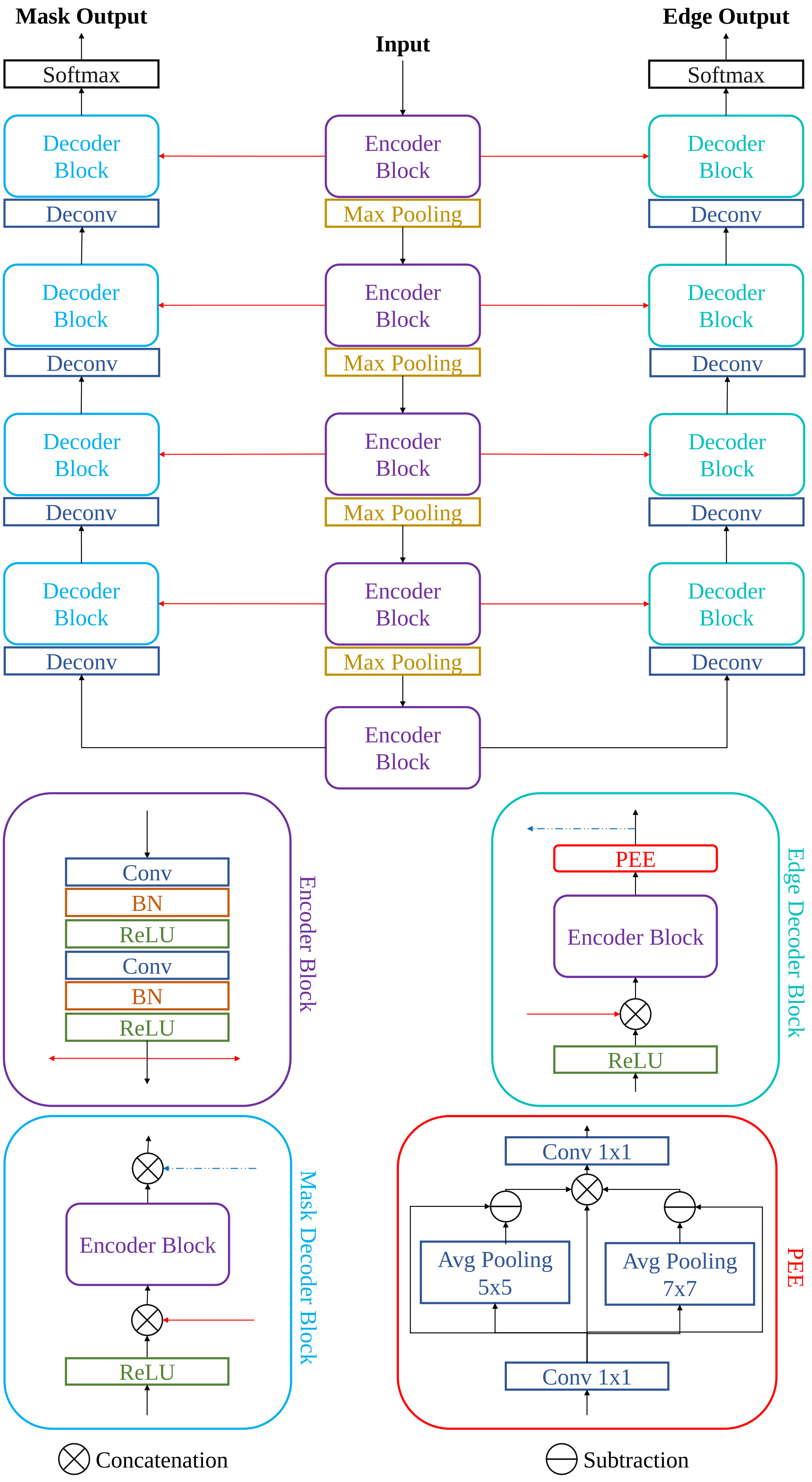


Figure 2.15: Graphical representation of CEL-UNet architecture.

2.6. Networks Training and Evaluation

Summarizing, in this work a total number of network architectures equal to five were considered:

1. *Ruthven-UNet*;
2. *QT-UNet*;
3. *CEL-UNet*;
4. *IM-UNet*;
5. *IM-UNet with Attention Block*.

Furthermore a total number of nineteen loss functions were tested:

1. *Cross-Entropy (CE) loss*;
2. *Weighted Cross-Entropy (WCE) loss*;
3. *TopK (TK) loss*;
4. *Weighted TopK (WTK) loss*;
5. *Focal loss (FL)*;
6. *Weighted Focal loss (WFL)*;
7. *Dice loss (DL)*;
8. *Dice-CrossEntropy (DCE) loss*;
9. *Dice-WeightedCrossEntropy (DWCE) loss*;
10. *Dice-TopK (DTopK) loss*;
11. *Dice-WeightedTopK (DWTopK) loss*;
12. *Dice-Focal (DF) loss*;
13. *Dice-WeightedFocal (DWF) loss*;
14. *Dice-CrossEntropy-TopK loss*;
15. *Dice-WeightedCrossEntropy-WeightedTopK loss*;
16. *Dice-CrossEntropy-Focal loss*;
17. *Dice-WeightedCrossEntropy-WeightedFocal loss*;

18. *Dice-CrossEntropy-Focal-TopK loss*;

19. *Dice-WeightedCrossEntropy-WeightedFocal-WeightedTopK loss*.

Since each architecture was trained and tested with all the 19 losses, 95 different networks were taken into account. They were all trained setting a batch size equal to 8, 70 epochs using as optimizer the ADAM one, a first-order gradient-based optimization of stochastic loss functions, with learning rate of 0.001. For each network the best one in term of the overall metric was saved and used to evaluate it on the test set. The project was developed in *Google Colab* environment using *TensorFlow v2.8.0*. The initial trainings were conducted exploiting *Google Colab* GPU NVIDIA Tesla T4 with RAM of 25 Gigabyte and each network took about one hour and 20 minutes to be trained. The successive trainings exploited a cluster of NVIDIA Tesla A100 with RAM of 40 Gigabyte each and each network took about 40 minutes to be trained. Particularly, the inference time for a single image was about 0.1s for QT-UNet, 0.084s for IM-UNet and 0.091s for IM-UNet with Attention Block.

2.6.1. Statistical tool

The results were analysed using the Kruskal-Wallis Test. It is the non parametric version of original one-way ANOVA. While the latter requires that the populations have normal distributions, the former requires only the mutual independence of all the observations. This test is based on the following hypothesis:
H_0: K independent groups that come from the same population and/or from populations having the same median;
H_1: K independent groups that don't come from the same population and/or from populations having the same median.
After having performed the Kruskal-Wallis test and founded the presence of global statistical differences, another tool was used to identify in a paired way the statistical significance between each group. This is called *Post hoc* test and, in this work, the one used was Tukey-Kramer test. All the analysis was performed in MATLAB R2021b environment.

2.6.2. Cross Validation

Cross Validation was applied on the best networks selected after the statistical analysis. The Subject-one-out Cross Validation was implemented using the 4 control subjects. Each network was trained and tested a number of times equal to the number of subjects, selecting them all cyclically. For each subject, the network was trained with a dataset including all subjects except the one selected, and tested with a dataset corresponding to the subject considered. This choice was made to guarantee that each subject could appear in both the training and the test set enhancing the variability of data.

2.7. Post processing

In order to reconstruct the complete segmentation of images and visualize it on 3D Slicer in *.nrrd* format, it was necessary to consider two problems. The first concerns the predictions fuzziness (continuous values between [0,1]) that must be converted in a crisp range 0,1, as the original manual segmented classes; this problem could have been solved by introducing thresholds, but it would have introduced problems of dubious membership of the pixels, as shown in the Figure 2.16(a). The problem of thresholding is in fact related to the output of the final network layer and led to the second problem which is the output probability map produced by the Soft-Max. This map indicated the probability of each pixels to belong to a given class and sometimes it could assign to the same pixel the probability of belonging to a class always lower than the threshold, for each class. This created the holes of unassigned pixels and was solved by searching for the highest probability that each pixel had among the classes, and by assigning it to the class to which this probability belonged (Argmax). This last problem was mainly related to the most uncertain areas, such as the edges.

This way the final segmentation has no more holes along borders among classes and can be robustly reconverted into *.nrrd* files, as can be seen in Figure 2.16(b).

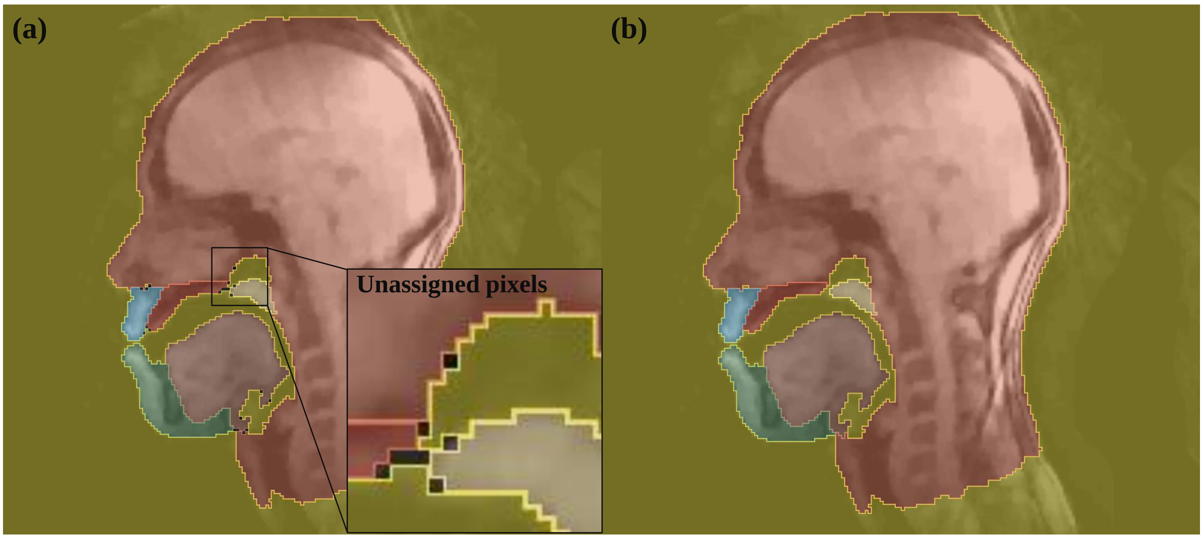


Figure 2.16: Example of: a) pre processed segmentation where there are holes along borders among classes, b) post processed segmentation where holes were assigned to one of the neighbouring classes, according to the highest probability of belonging to a class.

2.8. Vocal Tract Segmentation tool (VTS-tool)

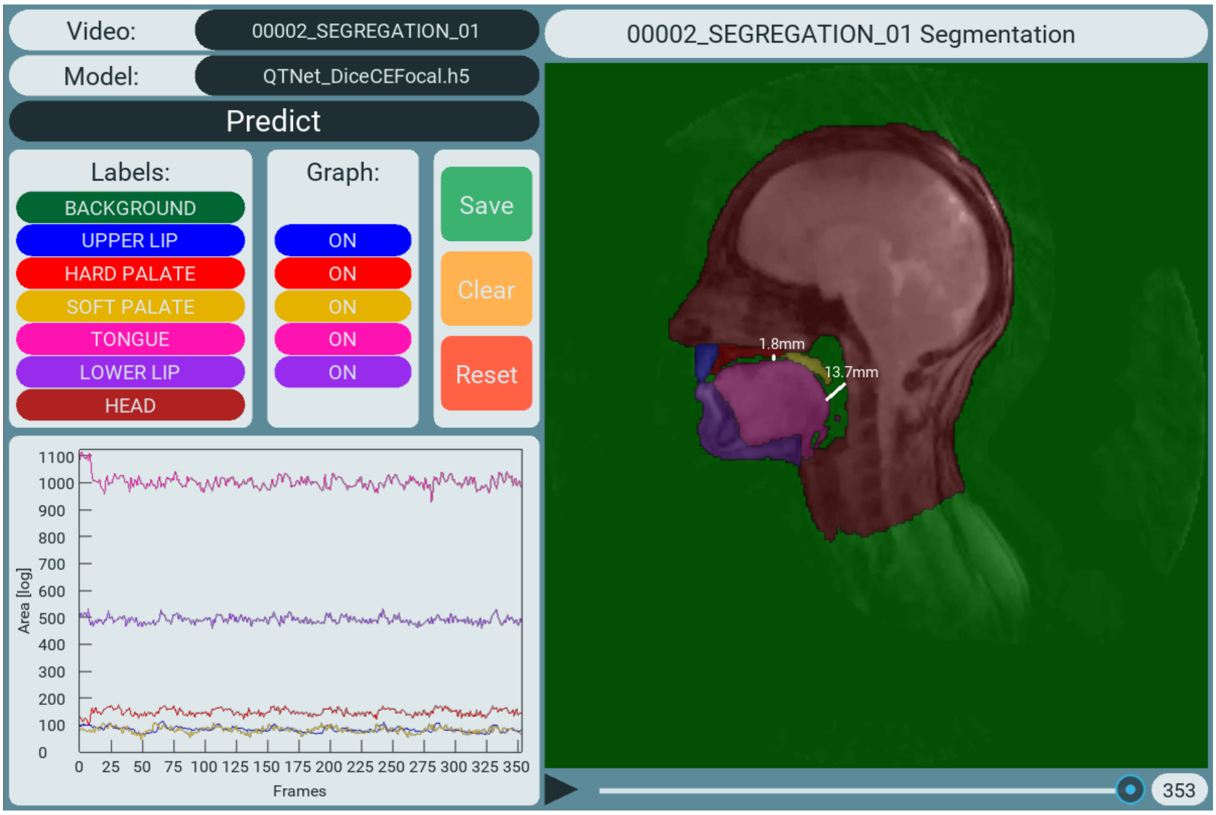
In order to make this work accessible to clinicians and allow them to benefit from the segmentations, in terms of timing, efficiency in recognizing areas and displaying data, a user friendly application was developed using Python 3.7.9 and KV Language. This VTS-tool allows clinicians to have a rapid segmentation of the MRI recording, reported in an interactive interface on which some useful operations and visualizations can be performed. Below there is an example of use cases:

- *video selection*: first, the selection of a video in (.avi) format of any MRI recording to be analyzed is required. The choice is made through a windows explorer;
- *network selection*: once a video is selected, it is necessary to select one of the three best networks obtained in this study, which will be used to perform the automatic segmentation;
- *prediction*: once the preliminary choices are made, by clicking on the *Predict* button, it is possible to start the automatic segmentation process. The selected video is divided into frames, which are segmented by the selected network. The areas

of articulators obtained are then post-processed with the method described in Section 2.7. Once the prediction is completed, the interactive area of the application is made usable;

- *labels*: it is possible to select the areas to be displayed, directly superposed on the frame shown. This section allows to highlight the movements of certain areas, clearly delineating their contours;
- *graph*: it shows the selected areas variation in real time. The concept of area variation would not make sense in a simple two-dimensional context as the areas should remain constant. In this case, however, they are areas of hard and soft tissues that expand in three dimensions, therefore the variation of the 2D area is strictly connected to the volumetric variation that occurs during speech. A decrease in area can mean a greater volumetric distribution on the z axis. This allows the user to recognize, at first glance, potential patterns that repeat themselves according to the task;
- *play button*: this button is used to sequentially play the frames, at the original video rate, so that moving areas can be viewed. The same effect can be recreated piecemeal by moving the slider;
- *drawing screen*: the portion of the screen where the various frames are depicted is interactive and allows the user to evaluate static distances on the active frame, also with labels active to highlight edges. These distances are shown in millimeters, knowing that the 3T Siemens Prisma scanner has a resolution of $1.6 \times 1.6\,mm^2$;
- *save*: areas raw data are saved in a (.csv) file for other possible applications, while their graphical trend into a (.pdf) file;
- *clear*: static distances projected on the screen are all cleared;
- *reset*: projected static distances, area segmentations and triggered trends are all returned to their initial state.

In conclusion, this application allows clinicians to perform rapid segmentation, to evaluate the trend of the areas and to recognize patterns, analyze the distances between areas in a static way and, eventually, be able to draw clinical conclusions.
Tool design is depicted in Figure 2.17.



Video: 00002_SEGREGATION_01

Model: QTNet_DiceCEFocal.h5

00002_SEGREGATION_01 Segmentation

Labels:	Graph:
BACKGROUND	
UPPER LIP	ON
HARD PALATE	ON
SOFT PALATE	ON
TONGUE	ON
LOWER LIP	ON
HEAD	

Figure 2.17: Application developed for predicted segmentation visualization, extraction of clinical metrics such as static distances and articulators areas.

3 | Results

3.1. Networks results

As described in Section 2.6, the 95 networks were trained and the evaluation on the test set was performed using the best version of each one, obtained across the epochs. The results, in terms of the three aforementioned metrics and the overall one, for all networks, are reported in the following twenty graphics that are divided according to the architectures. Results are shown as violin plots where each violin provides the distribution of metric values for all the test images, together with their box plot.

3.1.1. Dice

Dice metric trends of the networks are depicted in Figures 3.1 to 3.5. It can be noted that Dice values are all above the threshold of 0.6 with the only exception of CEL-UNet that has 1 network whose Dice metric is below 0.6. The three networks that are not reported in CEL-UNet graphic are CELUNet_WFocal, CELUNet_WTopK and CELUNet_DiceCE because they were considered as outliers, given the unmeaningful results that produced. It can be also noted that all networks built with compound loss functions have Dice values a bit higher with respect to networks built with single loss functions. Values of CEL-UNet networks are more stretched, meaning that performances of this network are pretty different among test images. Moreover, none of the distributions has normal shape.

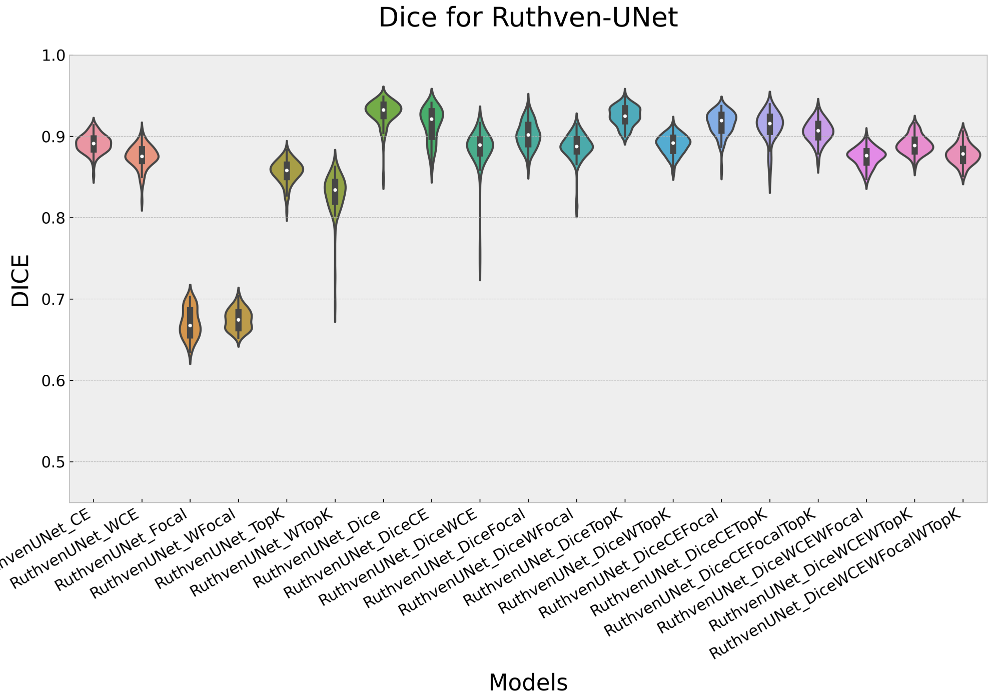


Figure 3.1: Violin Plot of Dice metric for networks built with Ruthven-UNet.

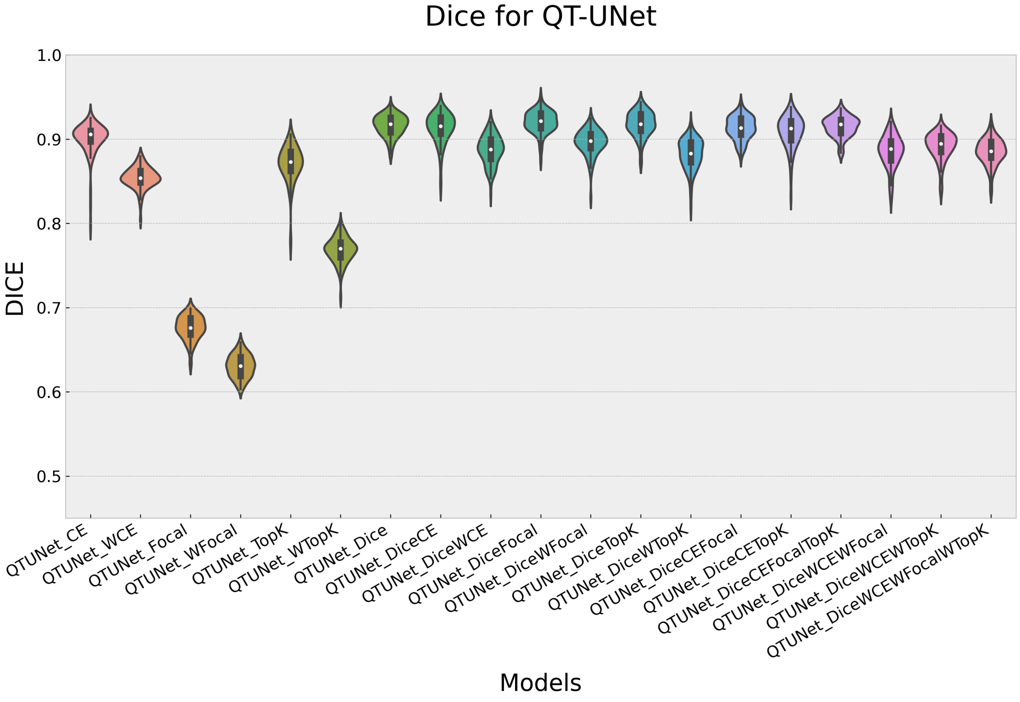


Figure 3.2: Violin Plot of Dice metric for networks built with QT-UNet.

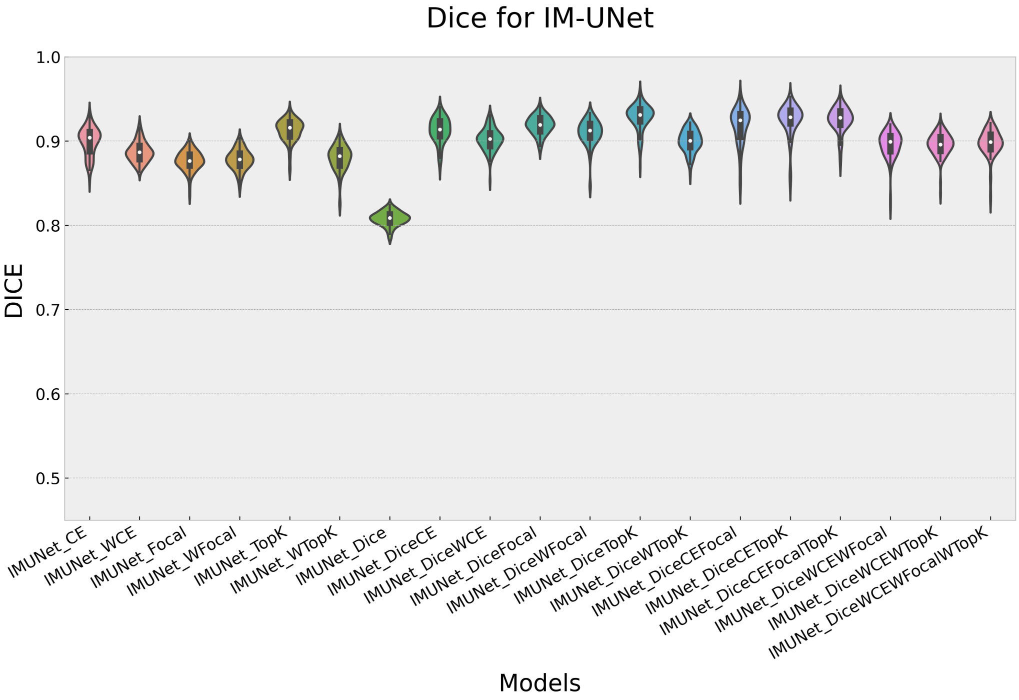


Figure 3.3: Violin Plot of Dice metric for networks built with IM-UNet.

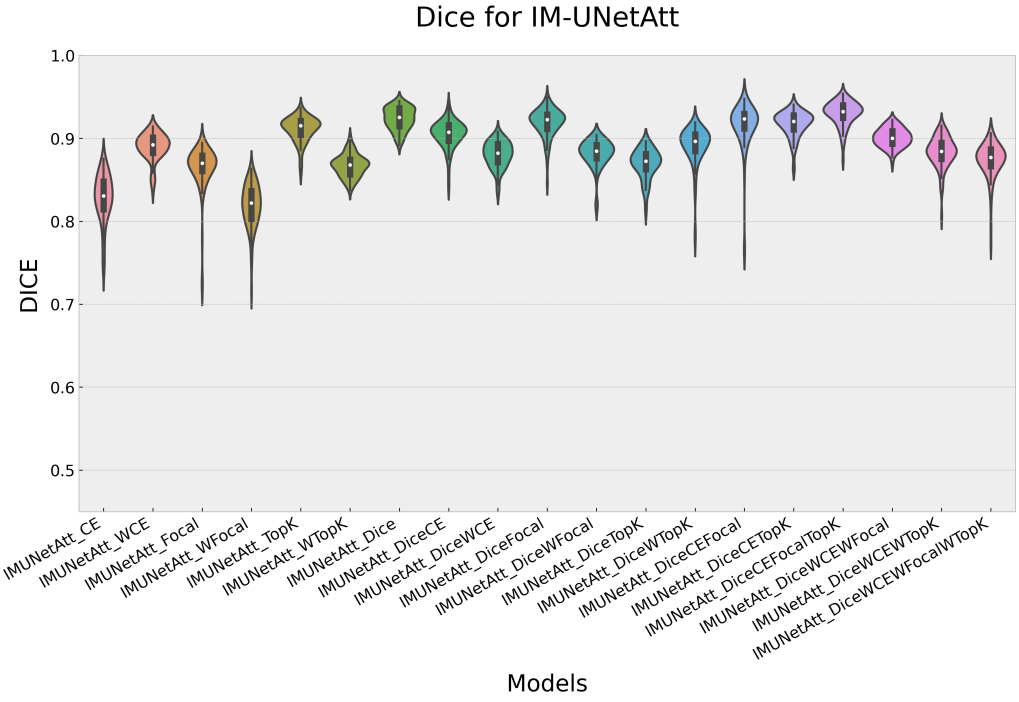


Figure 3.4: Violin Plot of Dice metric for networks built with IM-UNetAtt.

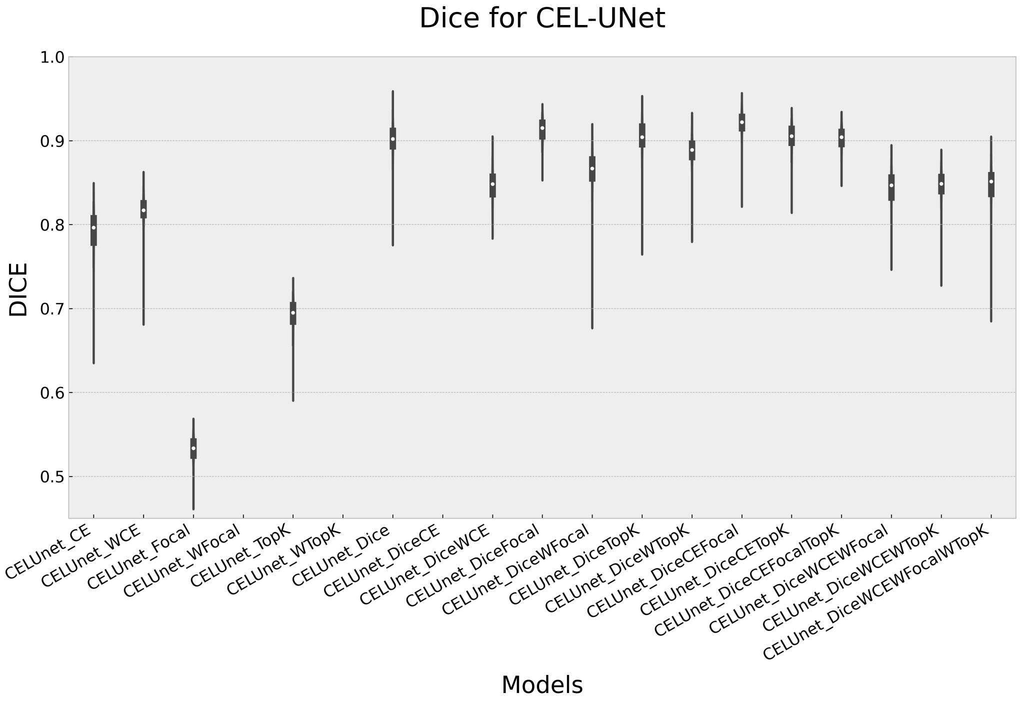


Figure 3.5: Violin Plot of Dice metric for networks built with CEL-UNet.

3.1.2. Hausdorff Distance

Hausdorff Distance metric trends are depicted in Figures 3.6 to 3.10, with the same conditions of Dice metric. It can be noted that the distributions are all between 0.2 and 0.45. For this metric, distributions of networks built with compound loss functions are comparable to the ones built with singular loss functions. The three outliers are again in the CEL-UNet graph, and again the violins are stretched, meaning that the performances of this network are pretty different among test images. Also for this metric distributions do not have normal shape.

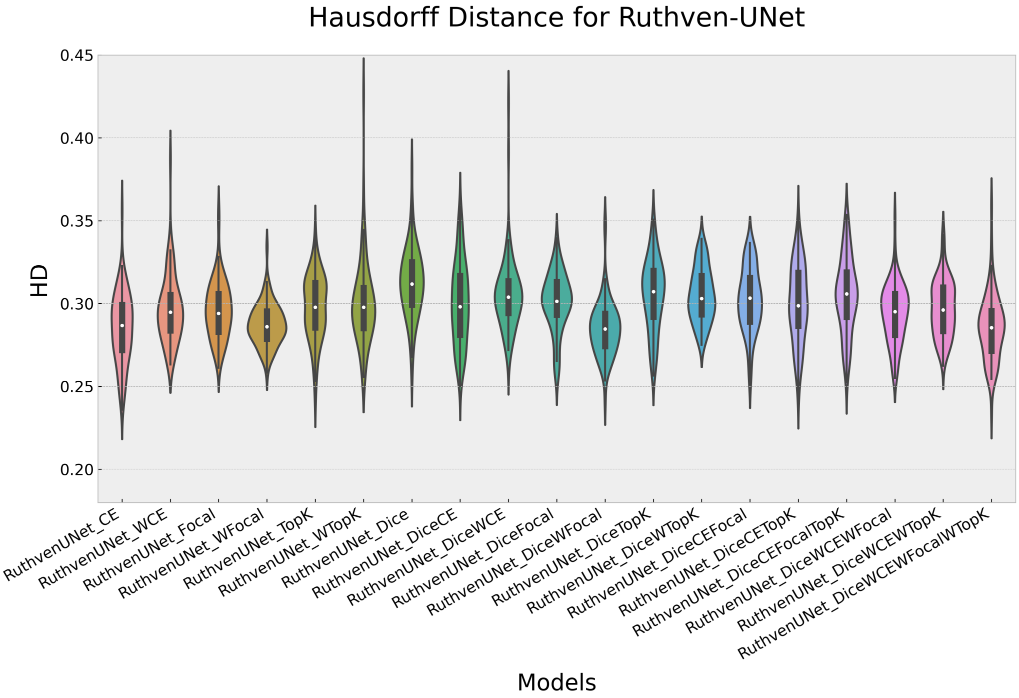


Figure 3.6: Violin Plot of Hausdorff Distance metric for networks built with Ruthven-UNet.

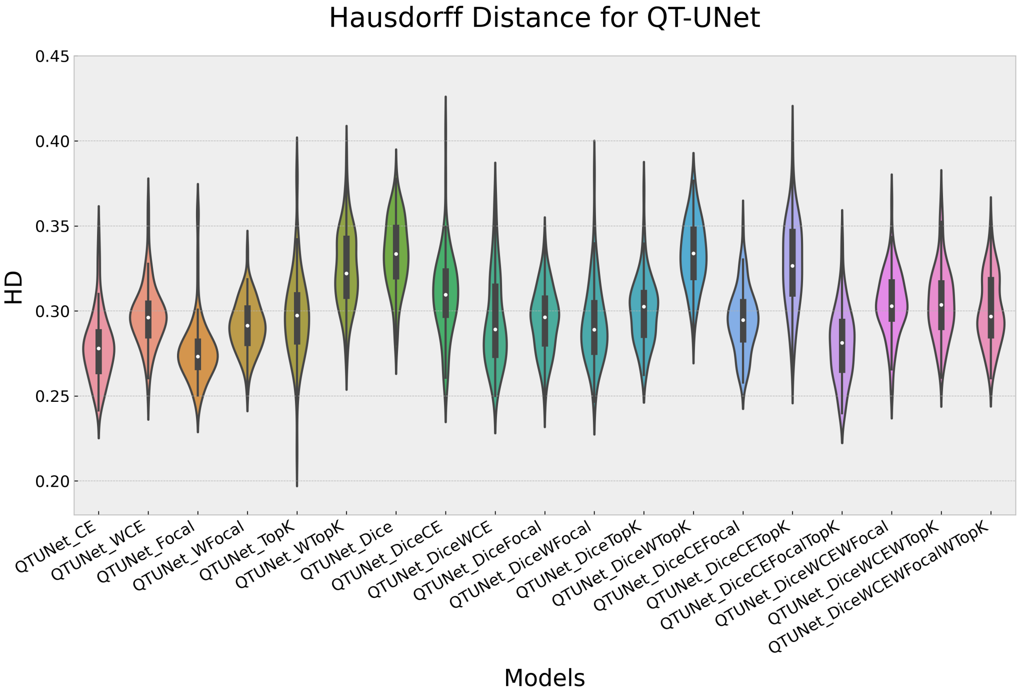


Figure 3.7: Violin Plot of Hausdorff Distance metric for networks built with QT-UNet.

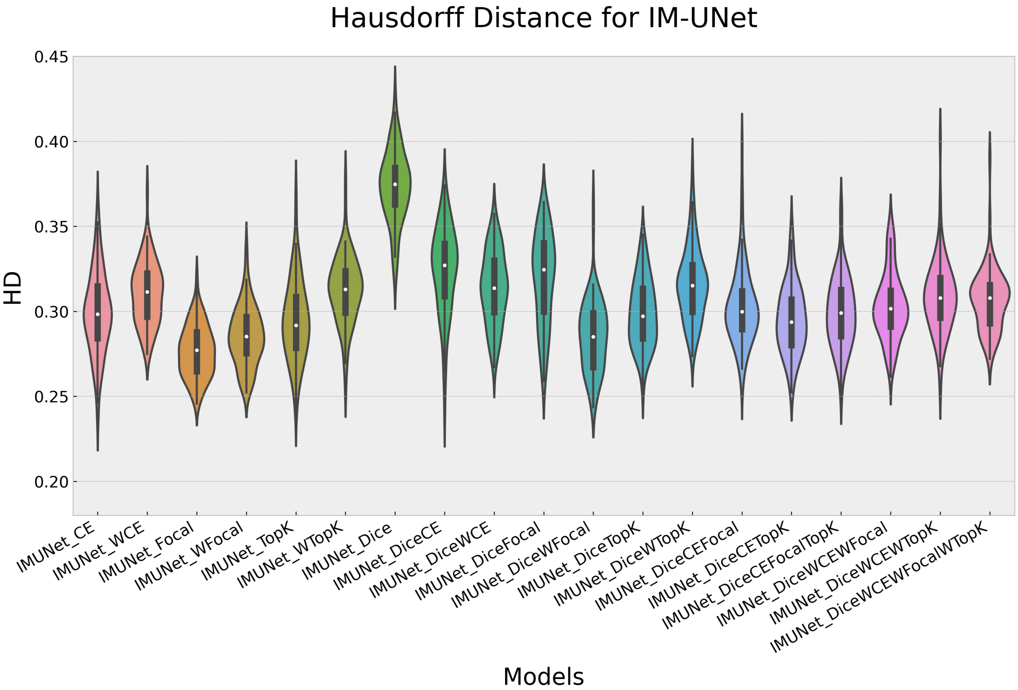


Figure 3.8: Violin Plot of Hausdorff Distance metric for networks built with IM-UNet.

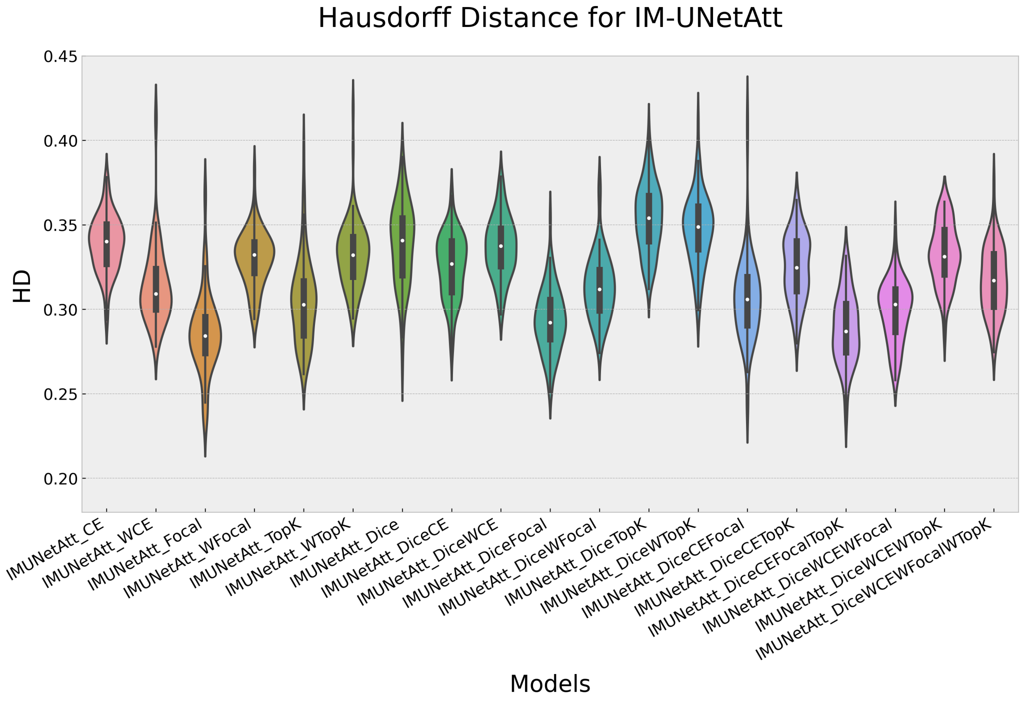


Figure 3.9: Violin Plot of Hausdorff Distance metric for networks built with IM-UNetAtt.

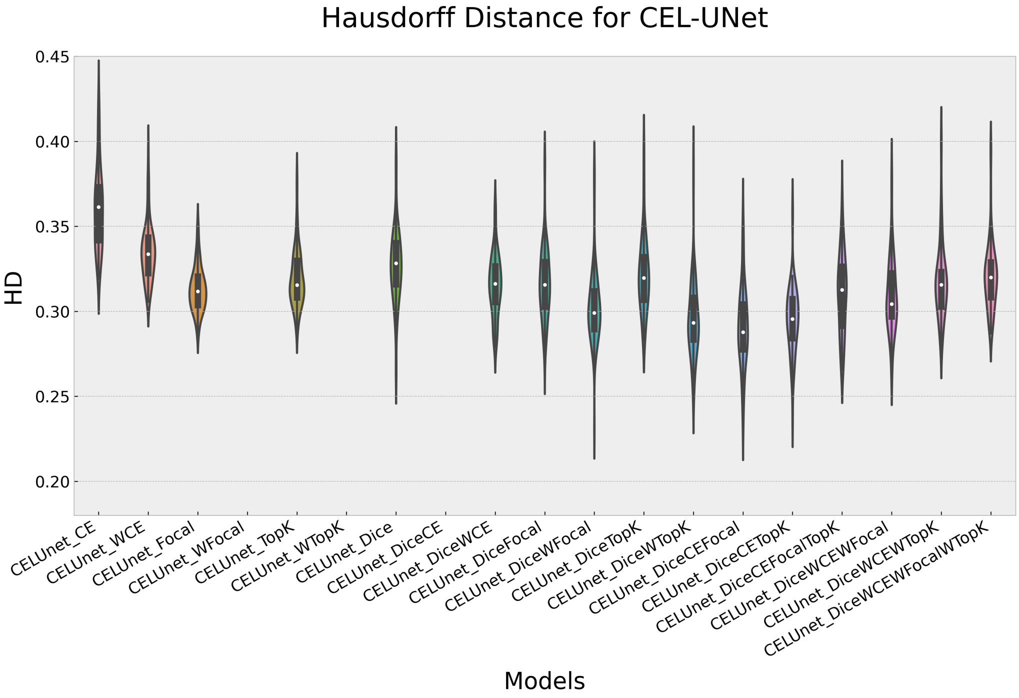


Figure 3.10: Violin Plot of Hausdorff Distance metric for networks built with CEL-UNet.

3.1.3. Global Consistency Error

Global Consistency Error trends are depicted in Figures 3.11 to 3.15, with the same conditions of the two previous metrics. It can be noted that the distributions are all between 0.0005 and 0.003. For this metric, it can be seen that distributions of networks built with compound loss functions are a little bit better (smaller) than the ones built with singular loss functions. CEL-UNet has again the three outliers that are not represented but, for this metric, the distributions are similar to the ones of the other architectures. Also in this metric distributions do not have normal shape.

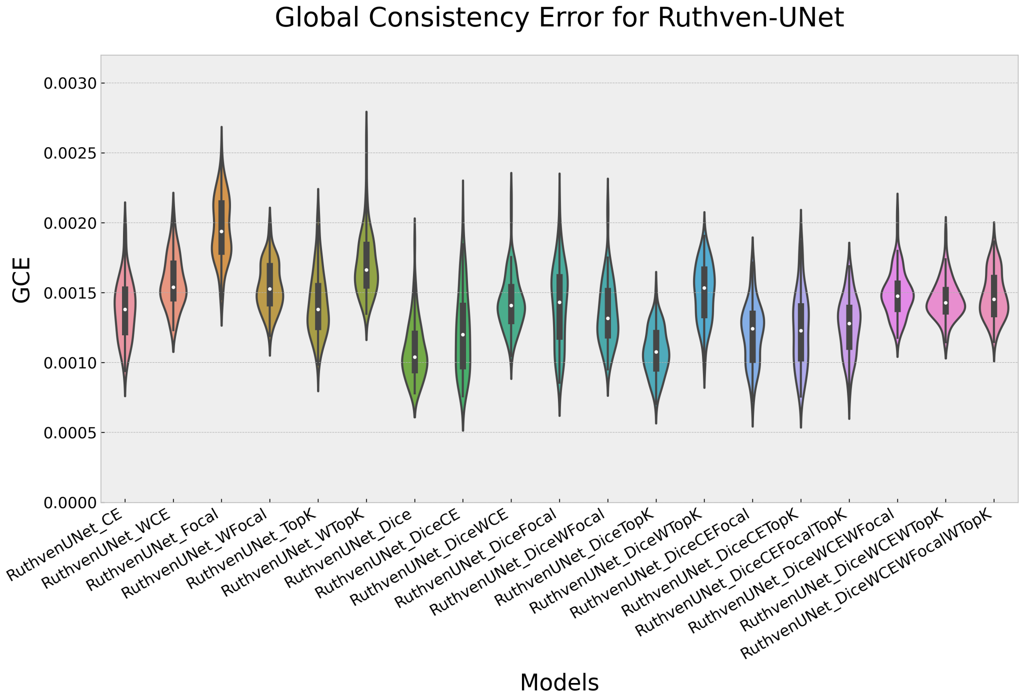


Figure 3.11: Violin Plot of Global Consistency Error metric for networks built with Ruthven-UNet.

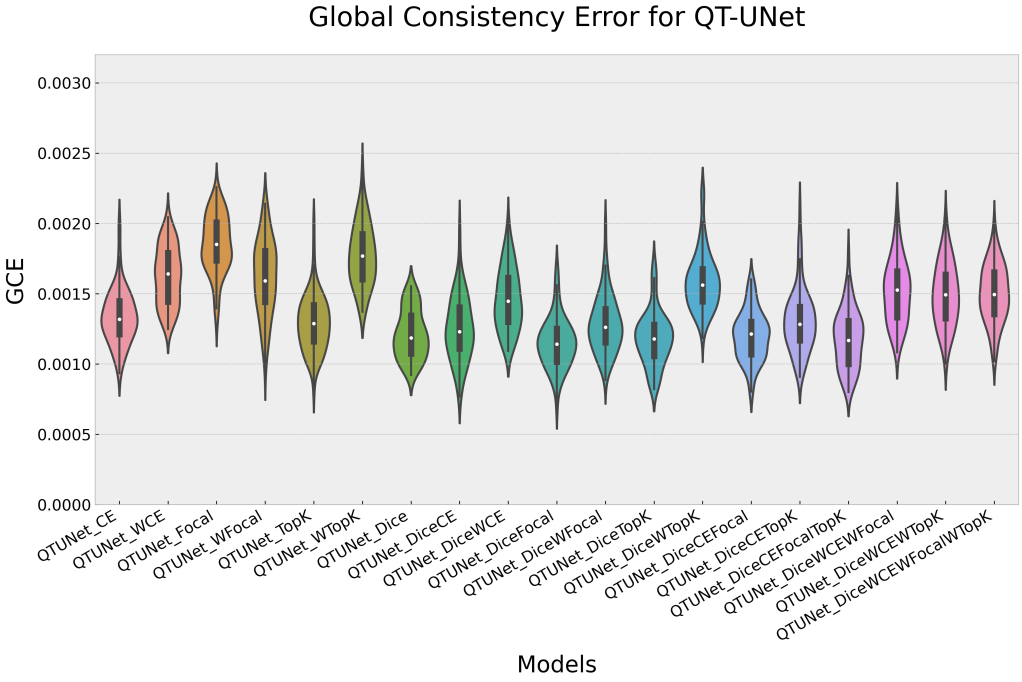


Figure 3.12: Violin Plot of Global Consistency Error metric for networks built with QT-UNet.

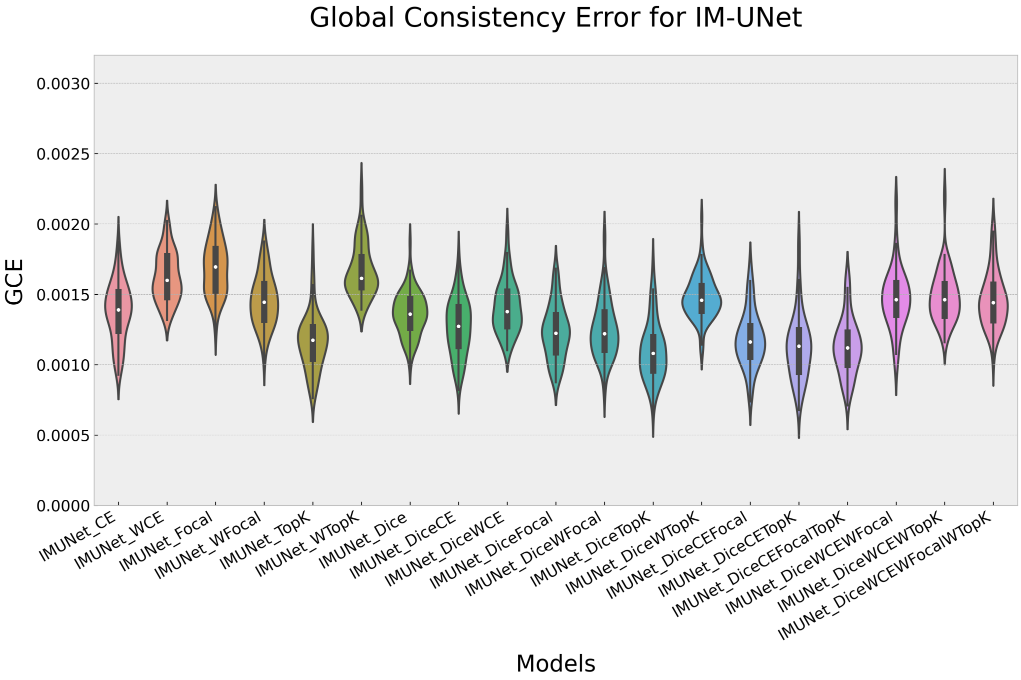


Figure 3.13: Violin Plot of Global Consistency Error metric for networks built with IM-UNet.

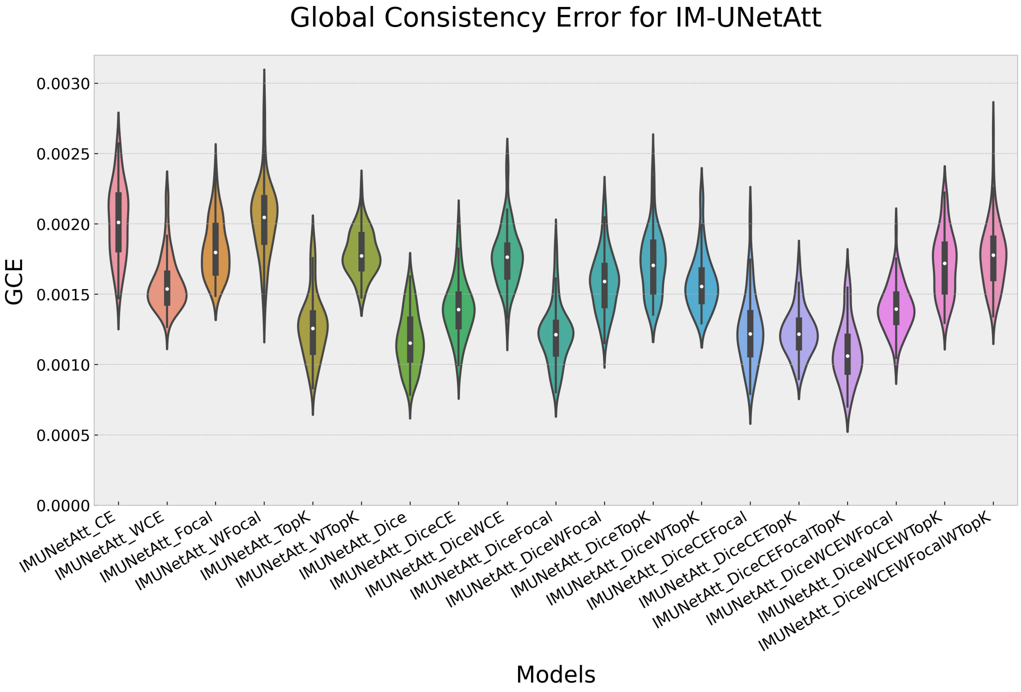


Figure 3.14: Violin Plot of Global Consistency Error metric for networks built with IM-UNetAtt.

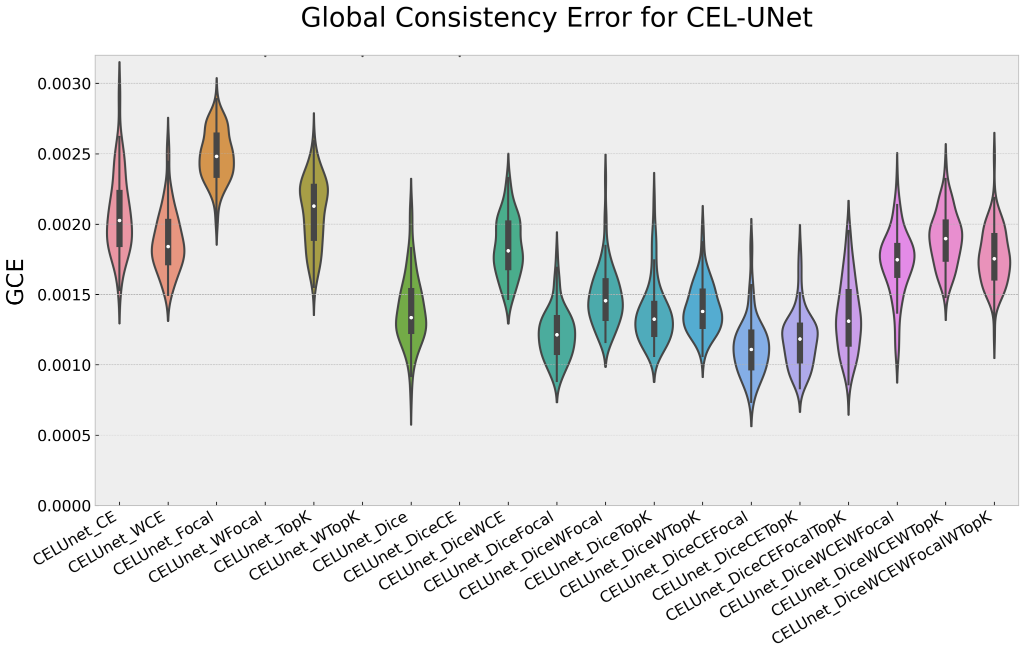


Figure 3.15: Violin Plot of Global Consistency Error metric for networks built with CEL-UNet.

3.1.4. Overall Metric

Lastly, overall metric trends are shown in Figures 3.16 to 3.20. Distributions are all below 0.8 threshold with the only exception of one CEL-UNet network. It has again the three outliers and very stretched distributions, meaning that the performances of this network are pretty different among test images. In general, networks built with compound loss functions gain quite lower values of OM with respect to the ones built with singular loss functions. Since this metric is given by a linear combination of the previous three it does not have a normal shape too.

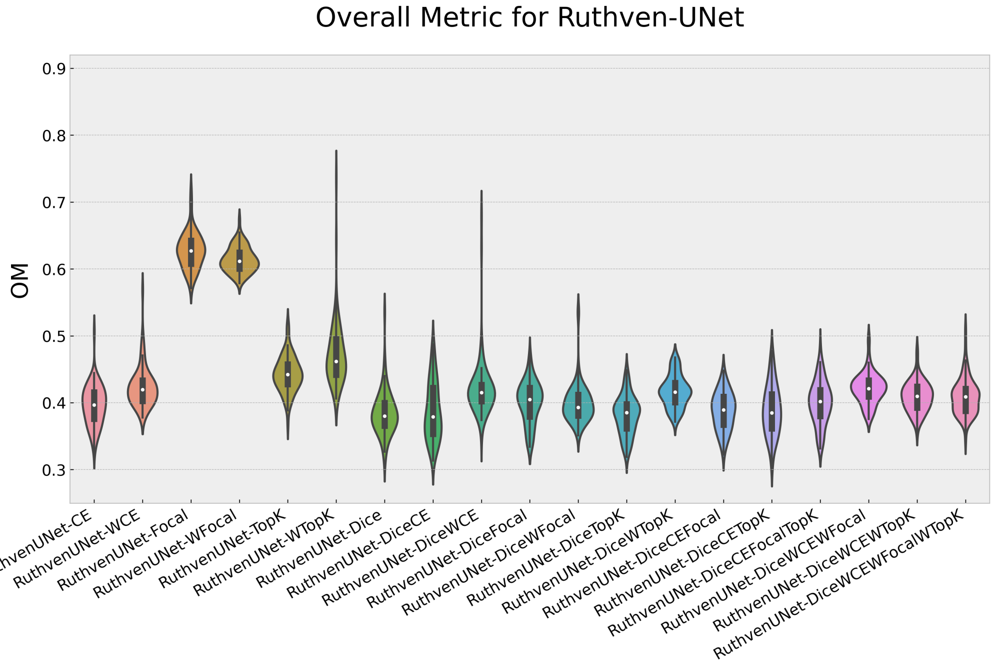


Figure 3.16: Violin Plot of overall metric for networks built with Ruthven-UNet.

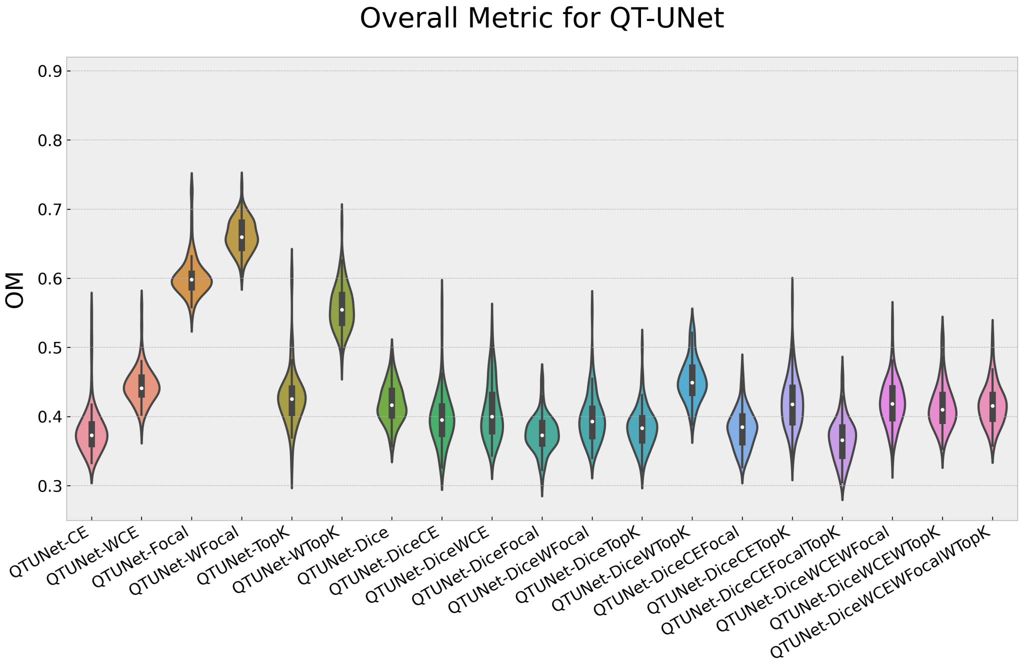


Figure 3.17: Violin Plot of overall metric for networks built with QT-UNet.

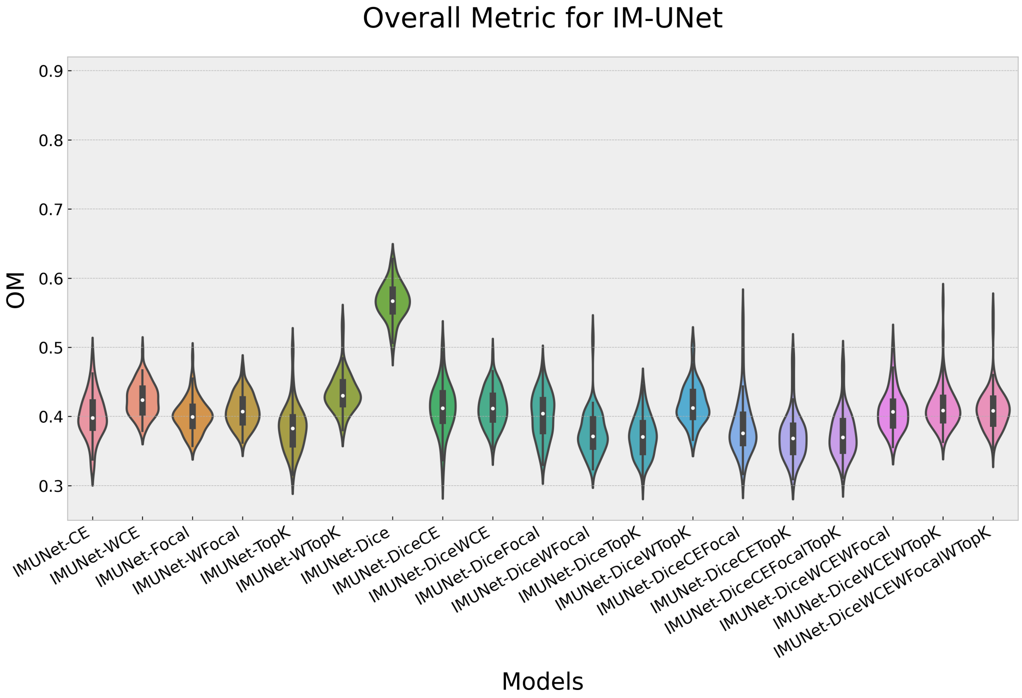


Figure 3.18: Violin Plot of overall metric for networks built with IM-UNet.

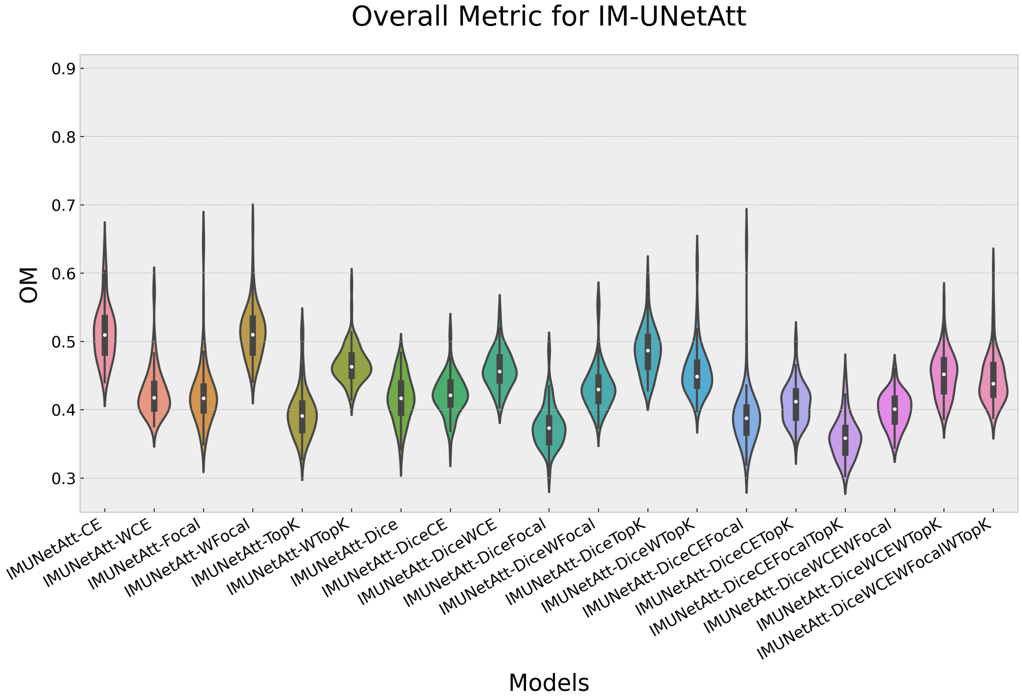


Figure 3.19: Violin Plot of overall metric for networks built with IM-UNetAtt.

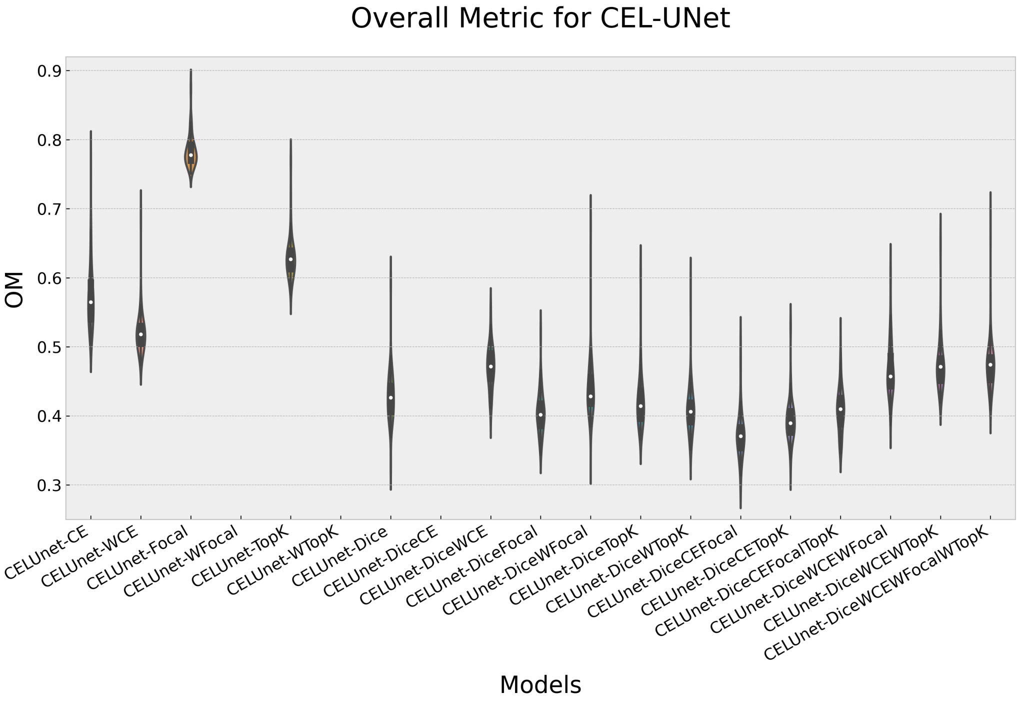


Figure 3.20: Violin Plot of overall metric for networks built with CEL-UNet.

3.2. Statistical Analysis

This section presents the results and considerations made to select the three best performing networks, with the related metric results.

3.2.1. Networks Ranking

In order to assert which are the best networks, it was necessary to conduct the statistical analysis described in Subsection 2.6.1. Kruskal Wallis Test was applied, and the groups compared were the 95 networks. Each group has 82 observations, which are the values of the overall metric for each image of the test set. The test produced a p-value equal to 0, lower than the significance level (fixed at 0.05), so the null hypothesis was rejected; at least two networks were significantly different then each other. In order to see which networks were or not significantly different from one another, the Tukey-Kramer test was conducted and the result is shown in Appendix A.1.

Selecting the network with the lowest median value as best overall, IM-UNet with Attention Block trained with *Dice-CrossEntropy-Focal-TopK loss*, there are 73 networks whose performance is significantly worse and 21 networks whose performance is not significantly different from it.

It was decided to first concentrate on the 22 best networks and see how they are distributed across both architectures and loss functions, obtaining a rank. As regards the five architectures:

- *CEL-UNet* belongs to 2 of the best networks;
- *IM-UNet with Attention Block* belongs to 4 of the best networks;
- *Ruthven-UNet* belongs to 5 of the best networks;
- *QT-UNet* belongs to 5 of the best networks;
- *IM-UNet* belongs to 6 of the best networks.

As regards the nineteen losses:

- *Dice loss, Cross-Entropy loss, Dice-CrossEntropy loss, Dice-WeightedFocal loss* belong to 1 of the best networks;
- *Dice-Focal loss, TopK loss* belong to 2 of the best networks;
- *Dice-TopK loss, Dice-CrossEntropy-TopK loss, Dice-CrossEntropy-Focal-TopK loss* belong to 3 of the best networks;
- *Dice-CrossEntropy-Focal loss* belongs to 5 of the best networks.

The remaining loss functions compose only networks which are significantly worse than the best ones.
Three networks were selected among the best 22 according to the following reasons:

- *IM-UNet with Attention Block* trained with *Dice-CrossEntropy-Focal-TopK loss* because it is the network with the lowest median value (1^{st} median);
- *IM-UNet* trained with *Dice-CrossEntropy-Focal loss* because it includes the architecture with the highest rank (6 networks) with the highest loss rank (5 networks) (1^{st} loss, 1^{st} architecture);
- *QT-UNet* trained with *Dice-CrossEntropy-Focal loss* because it includes the loss with the highest rank (5 networks) with one of the two architectures with the second highest rank (5 networks) (1^{st} loss, 2^{nd} architecture). QT-UNet was chosen compared to Ruthven-UNet because it has the lowest median value between the two.

The Post Hoc test produced a p-value equal to 1 when comparing IM-UNet trained with *Dice-CrossEntropy-Focal loss* and QT-UNet trained with *Dice-CrossEntropy-Focal loss*; a p-value equal to 0.9995 when comparing QT-UNet trained with *Dice-CrossEntropy-Focal* *loss* and IM-UNet with Attention Block trained with *Dice-CrossEntropy-Focal-TopK loss*; a p-value equal to 0.9763 when comparing IM-UNet trained with *Dice-CrossEntropy-Focal loss* and IM-UNet with Attention Block trained with *Dice-CrossEntropy-Focal-TopK loss*. These p-values are very elevated and quite similar to one another, meaning that these three networks are not statistically different from one another.

3.2.2. Best Networks Metrics

The numerical metrics results for these three networks are shown in Table 3.1. Since the distributions previously described are not normal, they are expressed in terms of median and interquartile range (IQR). They were computed on the 82 test set images taken from the four control subjects. As can be seen the IQR is low, compared to the median values, for all metrics in all networks. This means that the dispersion of values around the median is low.

Table 3.1: Best networks metrics results. They are expressed as median and interquartile range (IQR) and they are computed on the test set images taken from the four control subjects.

Network name	DICE	HD	GCE	OM
IM-UNet_DiceCEFocal	0.9248 (0.0271)	0.2997 (0.0223)	0.0012 (0.0002)	0.3756 (0.0412)
IM-UNetAtt_DiceCEFocalTopK	0.9321 (0.0152)	0.2869 (0.0287)	0.0011 (0.0002)	0.3583 (0.0362)
QT-UNet_DiceCEFocal	0.9134 (0.0192)	0.2947 (0.0218)	0.0012 (0.0002)	0.3847 (0.0373)

3.3. Cross Validation results

Then, Cross Validation, as described in Subsection 2.6.2, was performed and produced the metrics results shown in Table 3.2 expressed as median and IQR. It can be noted that median values are similar to the ones in Table 3.1; this means that the networks don't suffer from overfitting problem. IQR values instead are higher, meaning that values dispersion around median is greater.

Table 3.2: Cross Validation metrics results. They are expressed as median and interquartile range (IQR) and they are obtained performing the subject-one-out cross validation on the four control subjects.

Network name	DICE	HD	GCE	OM
IM-UNet_DiceCEFocal	0.9213 (0.0454)	0.2829 (0.0549)	0.0010 (0.0003)	0.3611 (0.0844)
QT-UNet_DiceCEFocal	0.9261 (0.0673)	0.2713 (0.0786)	0.0009 (0.0007)	0.3528 (0.1394)
IM-UNetAtt_DiceCEFocalTopK	0.8983 (0.0387)	0.3081 (0.0440)	0.0012 (0.0003)	0.4117 (0.0719)

3.4. Post processing results of control subjects

This section presents the metrics and the outputs of the segmentations relating to control subjects, obtained after post-processing.

3.4.1. Metrics results

Segmentations were reconstructed according to the procedure explained in Section 2.7 and they are evaluated with respect to the manual segmentations, producing the metrics results shown in Table 3.3. Results are expressed as median and IQR. As can be seen, all median values are a little bit better than the ones of pre processed images, shown in Table 3.1, while the IQR values are of the same order of magnitude.

Table 3.3: Best networks metrics results on post-processed images for control subjects. They are expressed as median and interquartile range (IQR) and they are obtained by the comparison between manual segmentations and the post processed segmentations.

Network name	DICE	HD	GCE	OM
IM-UNet_DiceCEFocal	0.9285 (0.0257)	0.3247 (0.0259)	0.0011 (0.0003)	0.3979 (0.0442)
IM-UNetAtt_DiceCEFocalTopK	0.9375 (0.0151)	0.3137 (0.0311)	0.0010 (0.0002)	0.3793 (0.0374)
QT-UNet_DiceCEFocal	0.9256 (0.0200)	0.3269 (0.0242)	0.0011 (0.0003)	0.4030 (0.0338)

3.4.2. Precision and Recall

The usual metrics together with Precision and Recall were used to evaluate the post processed segmentations divided according to classes. This was done in order to see if there were some differences in the performance of networks for different classes. The results are shown in Tables 3.4 to 3.8. As can be seen, precision results overcome 0.90 for almost all the metrics, this means that there are very few false positives. As regards recall instead, it can be seen that Soft Palate, Hard Palate and Lower Lip have values below or at most

equal to 0.90, meaning that the number of false negatives is higher than the one of the other four regions.
Regarding the Dice metric, it can be seen that Hard and Soft palates gain lower values with respect to the other regions; this difficulty in correctly classifying these regions can be found also in values of Hausdorff Distance metric, which are higher for those two regions. It must be noted that also Lower Lip region has this metric high.
Lastly, it can be seen that Global Consistency Error order of magnitude is directly proportional to the magnitude of the regions: larger regions, such as Background and Head, have values one order of magnitude bigger than medium regions, such as Lower Lip and Tongue, and two order of magnitude bigger than smaller regions, such as Upper Lip, Soft and Hard palates.

Table 3.4: Recall: best networks results on post-processed segmentations divided according to classes (Background, Upper Lip, Hard Palate, Soft Palate, Tongue and epiglottis, Lower Lip and Head). They are expressed as medians.

Network name	BK	UL	HP	SP	TO	LL	HE
IM-UNet_DiceCEFocal	0.9867	0.9046	0.8843	0.8254	0.9544	0.8568	0.9917
IM-UNetAtt_DiceCEFocalTopK	0.9927	0.9226	0.9125	0.8608	0.9451	0.8692	0.9722
QT-UNet_DiceCEFocal	0.9852	0.9554	0.8739	0.7897	0.9649	0.9076	0.9950

Table 3.5: Precision: best networks results on post-processed segmentations divided according to classes (Background, Upper Lip, Hard Palate, Soft Palate, Tongue and epiglottis, Lower Lip and Head). They are expressed as medians.

Network name	BK	UL	HP	SP	TO	LL	HE
IM-UNet_DiceCEFocal	0.9961	0.9705	0.9009	0.9071	0.9724	0.9776	0.9452
IM-UNetAtt_DiceCEFocalTopK	0.9917	0.9749	0.9169	0.9156	0.9863	0.9706	0.9688
QT-UNet_DiceCEFocal	0.9977	0.9274	0.8834	0.9056	0.9758	0.9345	0.9414

Table 3.6: Dice: best networks results on post-processed segmentations divided according to classes (Background, Upper Lip, Hard Palate, Soft Palate, Tongue and epiglottis, Lower Lip and Head). They are expressed as medians.

Network name	BK	UL	HP	SP	TO	LL	HE
IM-UNet_DiceCEFocal	0.9915	0.9269	0.8866	0.8583	0.9649	0.9144	0.9674
IM-UNetAtt_DiceCEFocalTopK	0.9921	0.9375	0.9086	0.8857	0.9655	0.9138	0.9694
QT-UNet_DiceCEFocal	0.9914	0.9350	0.8739	0.8478	0.9675	0.9186	0.9672

Table 3.7: Hausdorff Distance: best networks results on post-processed segmentations divided according to classes (Background, Upper Lip, Hard Palate, Soft Palate, Tongue and epiglottis, Lower Lip and Head). They are expressed as medians.

Network name	BK	UL	HP	SP	TO	LL	HE
IM-UNet_DiceCEFocal	0.2339	0.3015	0.3901	0.3780	0.3111	0.3849	0.3058
IM-UNetAtt_DiceCEFocalTopK	0.2421	0.2462	0.3901	0.3381	0.2840	0.4082	0.2682
QT-UNet_DiceCEFocal	0.2339	0.3015	0.3901	0.3381	0.2840	0.4303	0.2938

Table 3.8: Global Consistency Error: best networks results on post-processed segmentations divided according to classes (Background, Upper Lip, Hard Palate, Soft Palate, Tongue and epiglottis, Lower Lip and Head). They are expressed as medians.

Network name	BK	UL	HP	SP	TO	LL	HE
IM-UNet_DiceCEFocal	0.0261	0.0005	0.0011	0.0010	0.0024	0.0022	0.0235
IM-UNetAtt_DiceCEFocalTopK	0.0248	0.0004	0.0009	0.0008	0.0023	0.0022	0.0219
QT-UNet_DiceCEFocal	0.0263	0.0005	0.0012	0.0010	0.0021	0.0023	0.0238

3.4.3. Segmentation output

In Figure 3.21a there is an example of the ground truth segmentation of a control subject and in Figure 3.21b the correspondent predicted and post processed segmentation. As can be seen, these two segmentations reflect the results shown in Table 3.3.

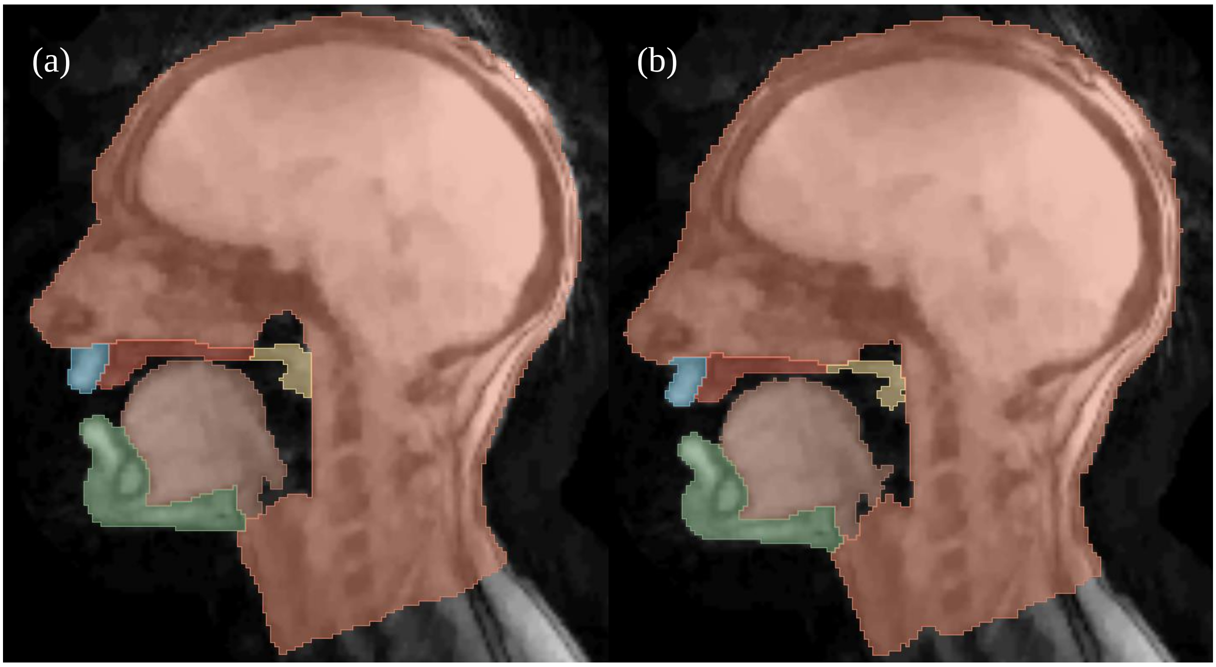


Figure 3.21: Example of a) ground truth segmentation of a control subject and b) the correspondent predicted and post processed segmentation.

In Figure 3.22, there is the example of a map showing, for a control subject, the ground truth area of regions with their edges (pink and red colours) with the superposition of areas that were under-segmented (purple) and over-segmented (green). Specifically, under-segmented areas are those ones that should have been included in a region but were not; over-segmented areas are those ones that should have been excluded from a region but were not.

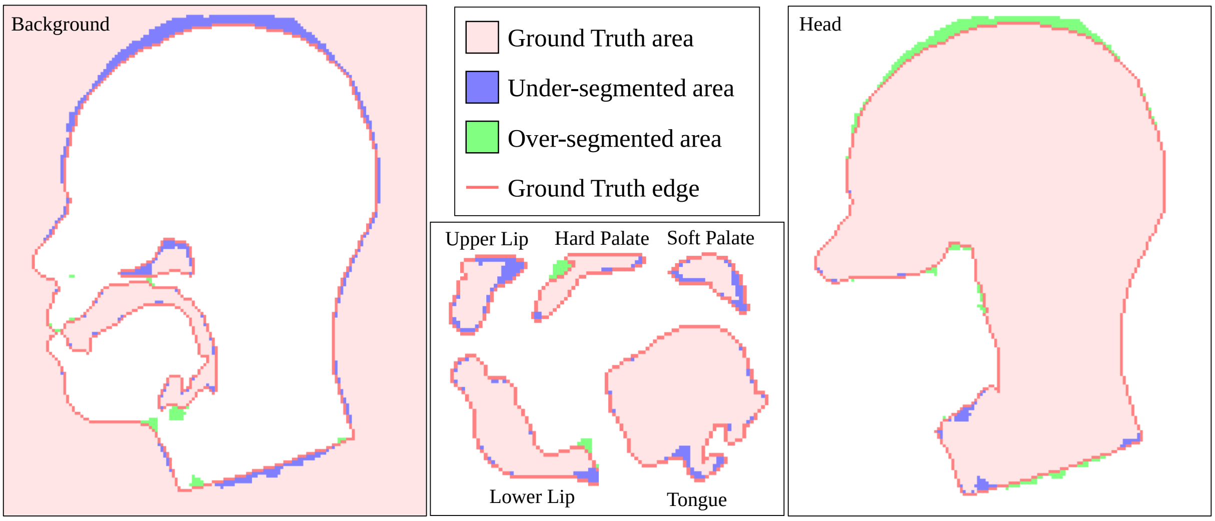


Figure 3.22: Example of a map showing, for a control subject, the ground truth area of regions with their edges (pink and red colours) with the superposition of areas that were under-segmented (purple) and over-segmented (green).

3.5. Post processing results of patient

As mentioned in Section 2.1, the best networks were also used to predict the segmentations of patient images, in order to test their generalizability.

3.5.1. Metrics results

Metrics results are synthesized in Table 3.9. They are again expressed as median and IQR. Taking into account that vocal structures of the patient are quite different from the control subject ones, metrics values get worse for all networks and have a IQR of the same order of magnitude of the one obtained after cross-validation.

Table 3.9: Best networks metrics results on post-processed images for patient. They are expressed as median and interquartile range (IQR) and they are obtained by testing the three best networks only on the images of the patient to test their generalizability.

Network name	DICE	HD	GCE	OM
IM-UNet_DiceCEFocal	0.8493 (0.0520)	0.3830 (0.0491)	0.0024 (0.0007)	0.5313 (0.1006)
IM-UNetAtt_DiceCEFocalTopK	0.8410 (0.0403)	0.3830 (0.0454)	0.0025 (0.0007)	0.5421 (0.0817)
QT-UNet_DiceCEFocal	0.7848 (0.0496)	0.3802 (0.0298)	0.0029 (0.0006)	0.5968 (0.0761)

The results distributions are also shown in four violin plots represented in Figures 3.23 to 3.26. It can be seen that for Dice metric, Global Consistency Error and overall metric the performances of IM-UNet with Attention Block trained with *Dice-CrossEntropy-Focal-TopK loss* and IM-UNet trained with *Dice-CrossEntropy-Focal loss* are slightly better than the ones of QT-UNet trained with *Dice-CrossEntropy-Focal loss*, while for Hausdorff Distance metric they are comparable.

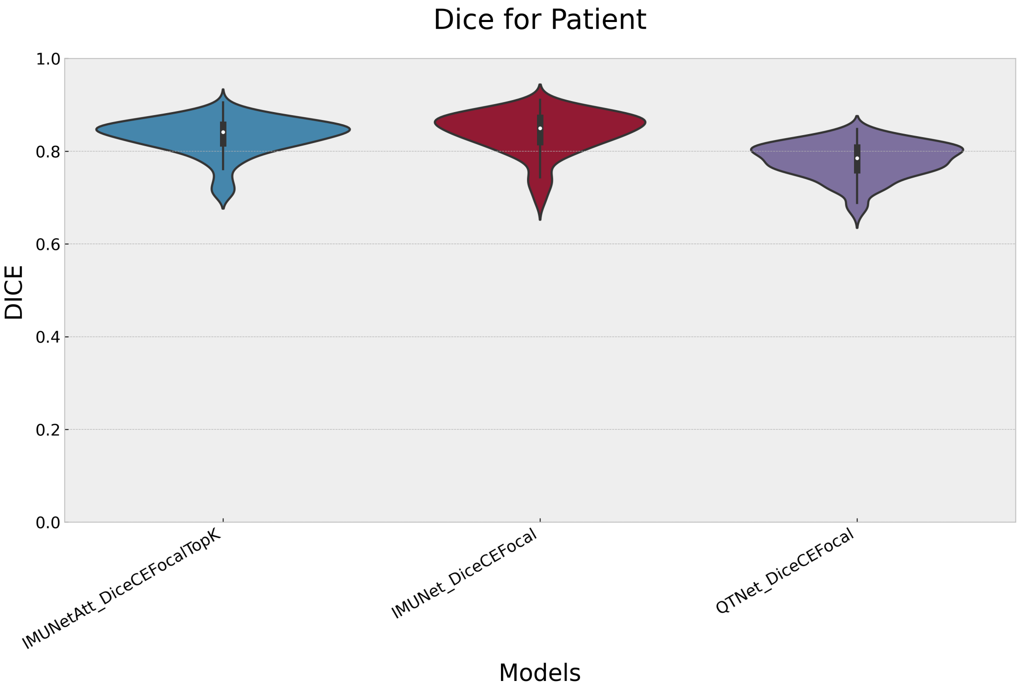


Figure 3.23: Violin Plot of Dice Metric for the three best networks tested on patient images.

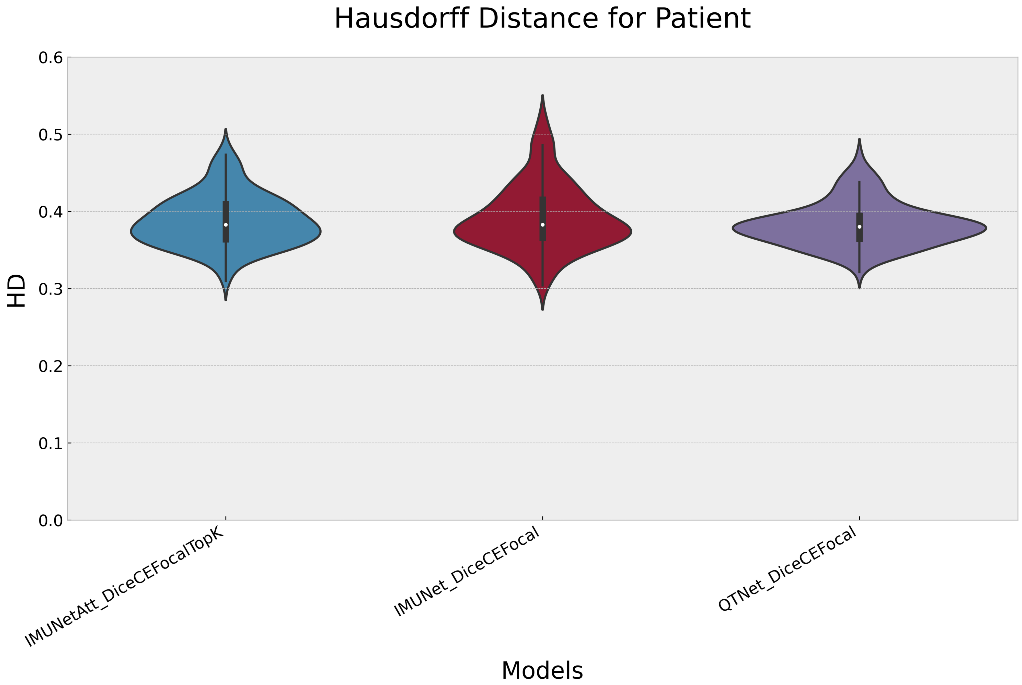


Figure 3.24: Violin Plot of Hausdorff Distance Metric for the three best networks tested on patient images.

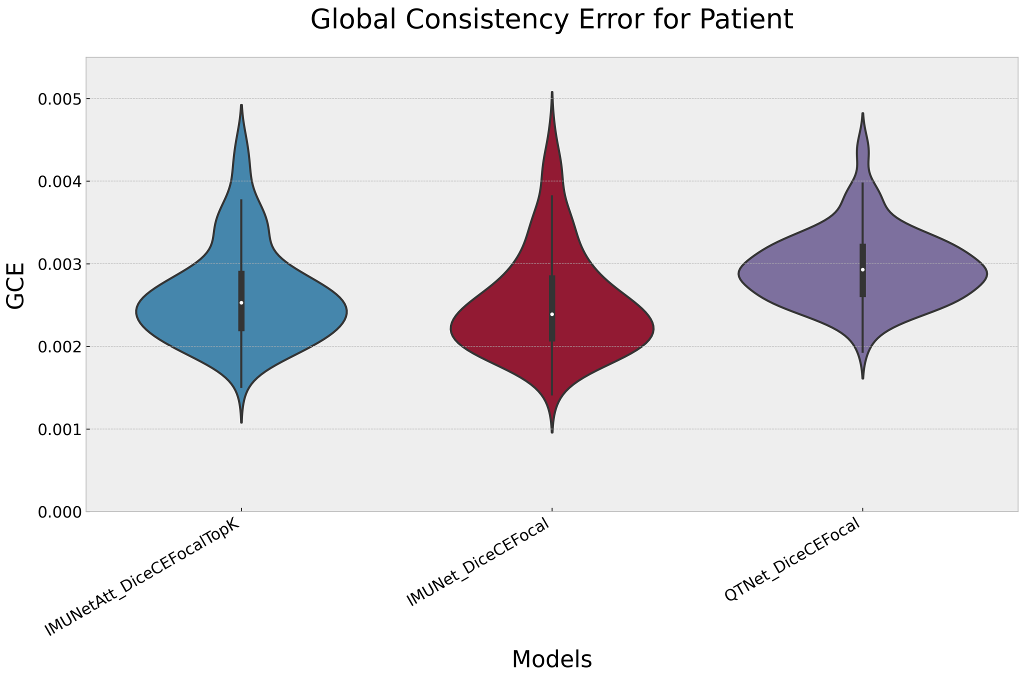


Figure 3.25: Violin Plot of Global Consistency Error Metric for the three best networks tested on patient images.

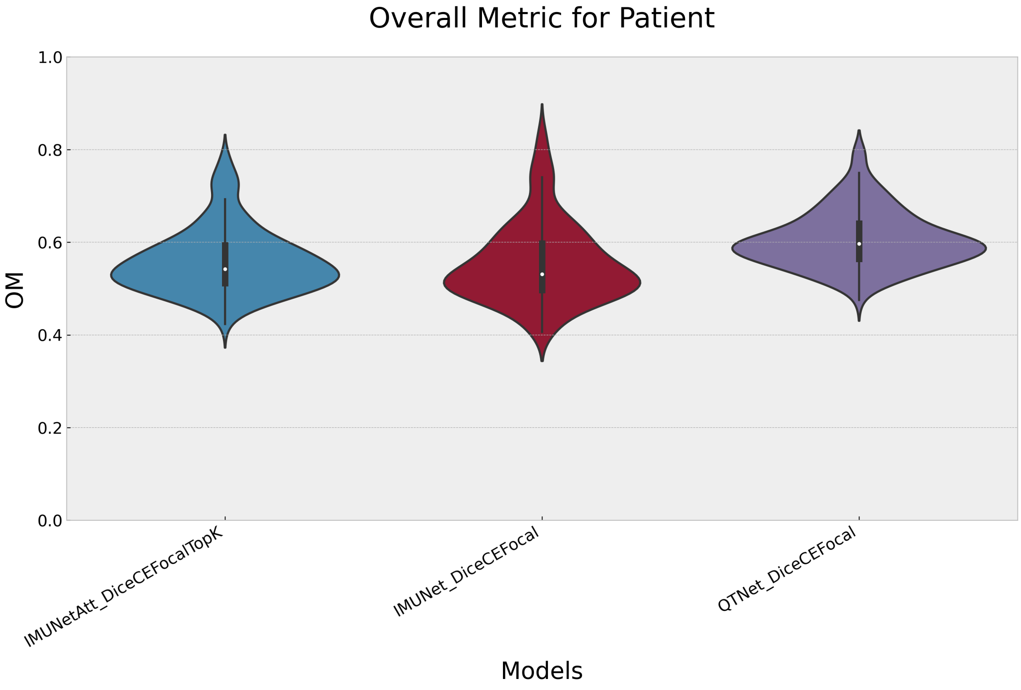


Figure 3.26: Violin Plot of overall metric for the three best networks tested on patient images.

3.5.2. Segmentation output

In Figure 3.27a there is an example of the ground truth segmentation of the patient and in Figure 3.27b the correspondent predicted and post processed segmentation. As can be seen, these two segmentations reflect the results shown in Table 3.9.

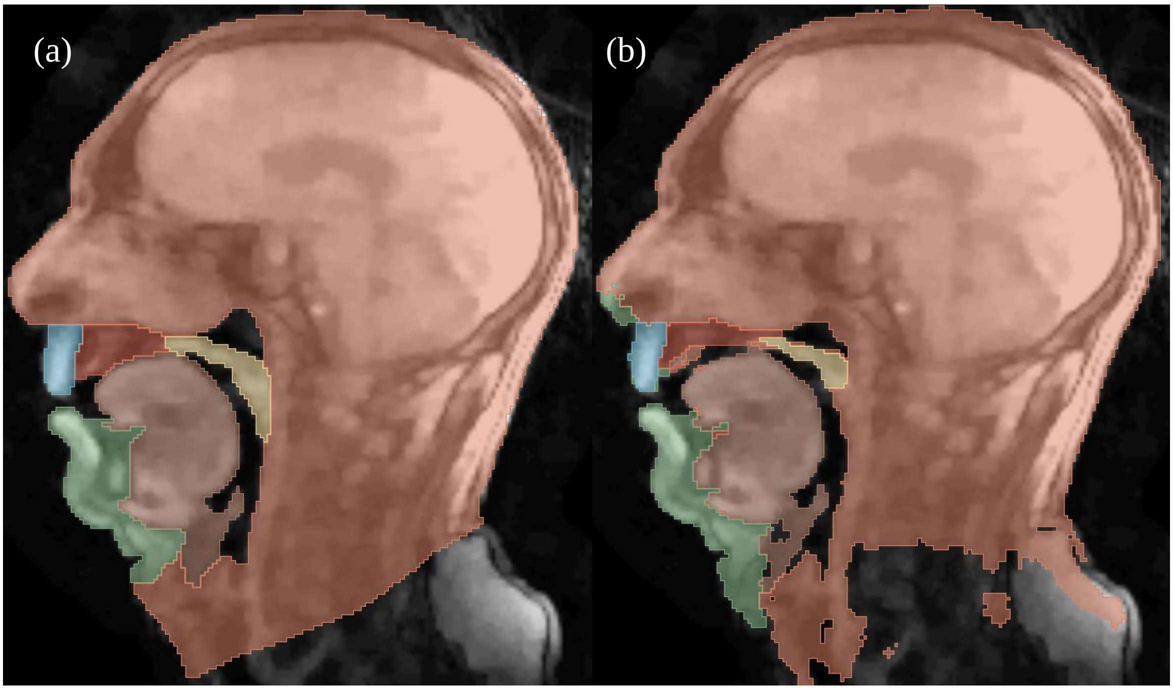


Figure 3.27: Example of a) ground truth segmentation of the patient and b) the correspondent predicted and post processed segmentation.

In Figure 3.28, there is the example of a map showing, for the patient, the ground truth area of regions with their edges (pink and red colours) with the superposition of areas that were under-segmented (purple) and over-segmented (green). Specifically, the under-segmented areas are those ones that should have been included in a region but were not; the over-segmented areas are those ones that should have been excluded from a region but were not.

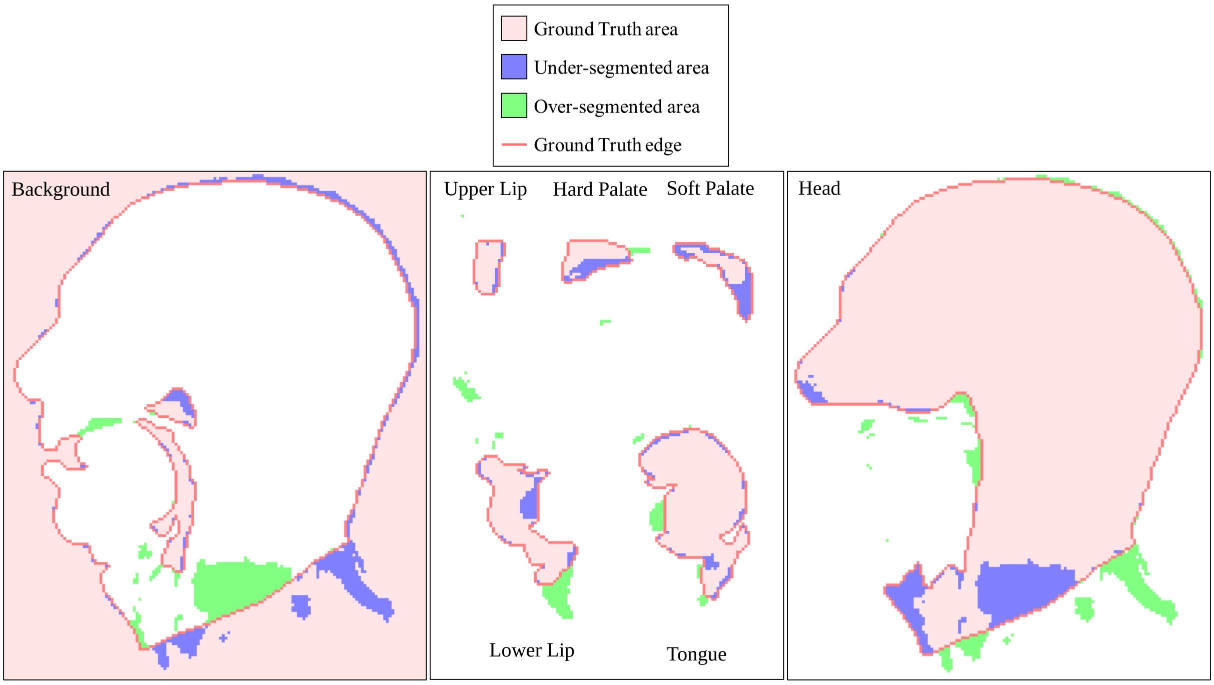


Figure 3.28: Example of a map showing, for the patient, the ground truth area of regions with their edges (pink and red colours) with the superposition of areas that were under-segmented (purple) and over-segmented (green).

3.6. Vocal Tract Segmentation tool results

As explained in Section 2.8, the VTS-tool provides the trend of the areas of articulators during the repetition of a task. Here below are graphically represented the trends of the articulators areas during the repetition of the task *Segregation*, obtained using QT-UNet trained with Dice-CrossEntropy-Focal loss. Figure 3.29 shows the areas trend for a control subject and Figure 3.30 for patient. Since each video is constituted by a total number of frames equal to 354, there are 354 values of articulators areas. Large areas such as head and background are not reported because their variation is pretty irrelevant. It can be seen that patient has a pattern that is significantly different from control for all the articulators considered.

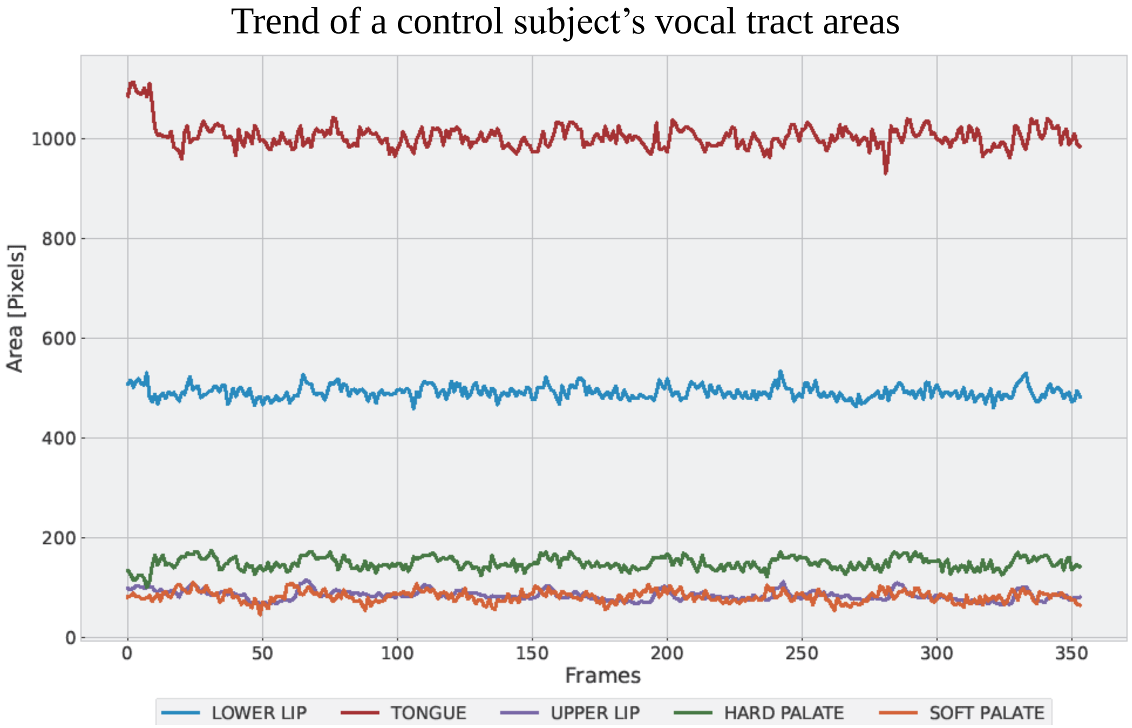


Figure 3.29: Trend of a control subject articulators areas during the repetition of the task *Segregation*. The number of considered frames is 354.

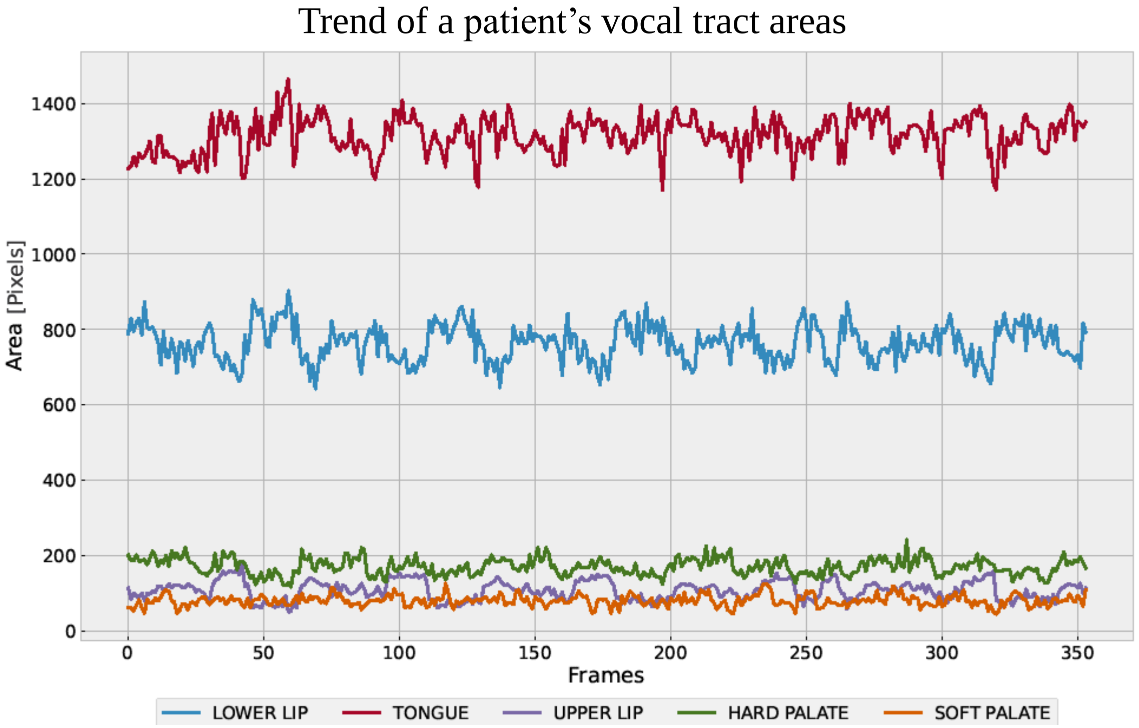


Figure 3.30: Trend of patient articulators areas during the repetition of the task *Segregation*. The number of considered frames is 354.

4 | Discussion

4.1. Loss functions results analysis

As described in Subsection 2.4.5, single loss functions at the pixels level seem to give rise to poorer performance due to their small value at the very beginning of the learning phase and the rapid decrease of the curve slope when approaching the end of the learning process. On the other hand, the more the loss functions are combined, the higher is the initial value and the slope, which is also kept until the best classification is reached. This allows the gradient to maintain a higher potential in the neighborhood of the correct segmentation, and therefore to be minimized with greater efficiency and precision.

From this evidence it is possible to suppose that compound loss functions learn better than single ones during the whole network learning, also at the topological level. This hypothesis is confirmed by the results obtained from the statistical analysis on the overall networks. In Section 3.2, loss functions were ranked according to their belonging to the best networks, identified by the Post Hoc test, and it can be seen that compound loss functions appear in the first positions. Moreover, it can be noted that combining more than two loss functions improves the performance of the networks, indeed the three best selected networks are composed by compound losses with three and four elements. This means that combining *Dice loss* and *Cross-Entropy loss* succeeds in taking into account the dissimilarities between the two distributions and also the overlap degree. Then, adding *Focal loss* allows to penalize the well-classified samples to focus on the worst ones and adding *TopK loss* allows to focus on the most difficult pixels, which are those points that the *Cross-Entropy loss* fails to classify.

4.2. Architectures results analysis

As described in Section 3.2, the three best networks are built with IMUNet, IMUNet with attention block and QTUNet architectures. The reason why these three architectures prevail over the RuthvenUNet and CELUNet may be linked to a common concept that they have in the encoding phase. This is the propagation of the initial information through all the layers before the *bottleneck*. The IMUNet (similarly to IMUNet with attention block), in its *encoding path*, propagates the initial information layer by layer, adding the linear projection of the input (1×1 Convolution) with a deeper convolution (3×3 Convolution). This way, information that is analyzed in a multi-scale level reaches the *bottleneck*. The QTUNet, instead, propagates the input to the *bottleneck* through the connections of the dense block, increasing the information flow received. Both architectures bring a portion of the initial information, and of each subsequent layer, to the *bottleneck*. It is therefore possible to state that it is preferable to propagate the input and subsequent layers, with residual or dense blocks, at the bottom of the network. This statement is consequently reflected on the *decoding path*, as it maintains the interconnections by propagating the deep knowledge even to the final layers.

This concept, as well as improving the quality of the network outputs, also improves its performances, as back-propagation is facilitated (mainly for QTUNet which has no operations in the interconnections, while IMUNet still has convolutions with weights to update).

In spite of the belonging of IMUNet with attention block to one of the best networks, it is necessary to notice that its performance is comparable to the one of the others two architectures. This means that the introduction of the attention block at the skip connection level does not significantly improve the performance of this network.

Furthermore, as regards IMUNet, replacing the convolutions with the transposed convolutions in the decoding processing block gave better results. Another aspect in common between the three networks is related to the bottleneck, which is not a classic convolution block as it is in RuthvenUNet and CELUNet. IMUNet, similarly to IMUNet with attention block, uses a series of dilated convolutions that keep the spatial resolution high even at the end of the process, however increasing the receptive field. QTUNet instead, uses a convolution (1×1) which linearly projects the information content coming from the *encoding path*, thus keeping its resolution constant too. Therefor, since the *bottleneck* does not have a max-pooling layer and therefore does not have the corresponding up-sampling layer that counterbalances it, it is preferable that it maintains the same resolution, without being subjected to classic convolutions with kernel higher than (1×1).

4.3. Best networks results analysis

As described in Section 3.2, the three best networks obtained are pretty equivalent in producing good results from a statistical point of view, since their p-values are very close to 1. Their performances, in terms of metric results are shown in Table 3.1. It can be noted that for all metrics the IQR is very low, meaning that the dispersion of values around the median is low and the median value is representative for the networks performance. Specifically, Dice values are all equal or greater than 0.91, meaning that the amount of overlap between the predicted classes and their ground truth is satisfactory. Hausdorff Distance values are quite close to 0, it means that the prediction's spatial position and boundaries are close to the ones of ground truth. Global Consistency Error values are very close to zero, meaning that also the overlap of the smallest portions of the regions is guaranteed and the amount of the true negatives is high as well as the amount of the true positives. Eventually, the overall metric values are quite close to 0 and it means that the overall performance of the networks can be considered satisfactory.

From Table 3.2, representing the subject-one-out cross validation results, it can be deduced that these networks don't suffer from overfitting problem because synthesis metrics values remain satisfactory, meaning that the networks are able to correctly segment images when trained and tested with different sets. Specifically, it means that these networks are able to well predict structures of a subject after a training phase conducted on all the other subjects. So their ability to predict something new can be considered satisfactory. Obviously the IQRs are a little bit higher than the ones of Table 3.1 because the variability of values obtained with cross validation procedure is stronger.

4.4. Patient results analysis

The evaluation of the best networks on patient segmentations was performed to test their generalizability. As can be seen in Figure 3.27 the vocal tract structures of the patient are in quite a different shape compared to a control subject, and because of that it was more difficult for networks to perform a correct segmentation. In fact there are several groups of pixels belonging to the incorrect region, especially in the lowest portion of the head or at the borders among Upper Lip, Hard Palate and Soft Palate. However, the results shown in Table 3.9 demonstrate that the three networks perform quite well for all metrics. Their IQR is higher compared to the one of the networks tested on control subjects, this means that there is more variability in predicted segmentations. But IQR values are small enough not to compromise the representativeness of the median values.

In Figure 3.28 can be seen the over and under segmented areas. We would expect a predominance of under-segmented areas since the Focal loss function used in all networks tends to provide a more conservative segmentation. But, since performances for the patient are not so good, this tendency does not emerge.

4.5. Post processing results analysis

A can be seen in Table 3.3, the post processing of predicted segmentations improves a bit the metrics results. This happens because all pixels are assigned to at least one class, according to the highest probability of belonging to that class. This way there are no more holes along borders between regions and the reconstructed overall segmentation is more similar to manual one, as can be seen in Figure 3.21.

The last step was analyzing Precision, Recall, Dice metric, Hausdorff Distance metric and Global Consistency Error metric, separately for each region, to see if there are some differences between them. As can be noted from Table 3.4 and Table 3.5, the values of Precision are always greater than those of Recall, this is caused by the use of Focal loss function in all networks. This loss function amplifies Precision to the detriment of Recall, meaning that it privileges a sub-segmentation rather than over-segmentation. This tendency is confirmed by the map shown in Figure 3.22, where there is the strong predominance of under-segmented areas (purple ones). In particular this is evident in proportion for Hard and Soft palate, that are small and highly variable regions. In fact, as can be seen in Tables 3.6 to 3.8, these two regions gain worse values also for the other three metrics with respect to the other regions. In general these results confirm that small and highly variable regions, such as hard and soft palate, are more challenging to be correctly segmented, while big and low variable regions, such as head and background, are easier to be correctly segmented. Eventually, upper lip gains quite good results because it is a small region but it does not have great variability; at the opposite, tongue and lower lip are medium regions but their variability is significant. This emerges from their results which are worse than those of head and background, but better than those of hard and soft palate.

4.6. Vocal Tract Segmentation tool results analysis

The computation of articulators area and the visualization of their trend in time is one of the possible clinical metric that can be used to discriminate between a physiological production of speech and the presence of some motor speech impairment. In particular, as explained in Section 0.1, apraxia of speech (AOS) is characterized by inconsistent speech patterns, while dysarthria is connoted by consistent patterns. This way, by looking at the aformentioned trends, is possible, for example, to discriminate between these two pathological conditions. Figure 3.29 and Figure 3.30 provide an example of articulators area trends representing a physiological and a pathological condition.

5 | Conclusion and future development

The automatic segmentation of vocal tract in its main articulators was successfully performed by the best networks obtained. They were all built with compound loss functions made by three or four losses, proving the superiority of multiple losses with respect to double or singular ones. They were all built with architectures that improved *bottleneck* information flow (dense blocks or residual blocks), and *bottlenecks* that maintain spatial resolution.

The best networks gained satisfactory metrics results on control subjects and good results on patient. The patient results and the cross validation results also demonstrate that these networks achieve quite good generalizability and don't suffer from overfitting problem. Moreover, the post processing procedure succeeds in reconstructing the whole predicted segmentation by removing holes along articulators borders and achieving slightly better metrics results. The VTS-tool developed allows to visualize, in a dynamical manner, the MRI images with their predicted segmentations superposed and it gives the possibility to the clinicians to compute some clinical metrics both interactively (distances) and automatically (dynamic computation of articulators areas). This VTS-tool allows clinicians to save time, because it is not necessary to perform the manual segmentation of each dsMRI image, which is a very time-consuming activity. It also allows clinicians to obtain quantifiable and objective clinical information that can help them making an early diagnosis and a better monitoring of speech diseases.

The main limitation of this work is the small amount of subjects included in the dataset which leaded to patient results that are acceptable but not sufficient to help significantly clinicians. To improve predictions of patients articulators it will be necessary to increase significantly the number of controls and patients included in the dataset in order to obtain higher values of generalizability. Another future development could concern the VTS-tool that can be customized according to the specific clinical metrics requested by clinicians. For example, pattern-recognition networks can be developed to quantitatively define the presence of recurring patterns in the trend of the areas. Eventually, it could be useful to create a predictive model for a dynamic reference system, which can be used by the VTS-tool to determine the distances of the vocal tract in a dynamical manner.

Bibliography

[1] C. P. Browman and L. Goldstein, "Articulatory Phonology: An Overview," *Phonetica*, vol. 49, no. 3-4, pp. 155–180, May 1992. DOI: 10.1159/000261913.

[2] S. Brown, E. Ngan, and M. Liotti, "A Larynx Area in the Human Motor Cortex," *Cerebral Cortex*, vol. 18, no. 4, pp. 837–845, Apr. 2008. DOI: 10.1093/cercor/bhm131.

[3] J. R. Duffy, *Motor speech disorders e-book: Substrates, differential diagnosis, and management*. Elsevier Health Sciences, 2019, ISBN: 9780323550512.

[4] C. L. Ludlow, N. P. Connor, and C. J. Bassich, "Speech timing in Parkinson's and Huntington's disease," *Brain and Language*, vol. 32, no. 2, pp. 195–214, Nov. 1987. DOI: 10.1016/0093-934X(87)90124-6.

[5] K. A. Josephs, "Clinicopathological and imaging correlates of progressive aphasia and apraxia of speech," *Brain*, vol. 129, no. 6, pp. 1385–1398, Apr. 2006. DOI: 10.1093/brain/awl078.

[6] M. Grossman, "Biomarkers to Identify the Pathological Basis for Frontotemporal Lobar Degeneration," *Journal of Molecular Neuroscience*, vol. 45, no. 3, pp. 366–371, Nov. 2011. DOI: 10.1007/s12031-011-9597-0.

[7] F. Caso, M. L. Mandelli, *et al.*, "In vivo signatures of nonfluent/agrammatic primary progressive aphasia caused by FTLD pathology," *Neurology*, vol. 82, no. 3, pp. 239–247, Jan. 2014. DOI: 10.1212/WNL.0000000000000031.

[8] J. M. Ogar, N. F. Dronkers, *et al.*, "Progressive Nonfluent Aphasia and Its Characteristic Motor Speech Deficits," *Alzheimer Disease & Associated Disorders*, vol. 21, no. 4, S23–S30, Oct. 2007. DOI: 10.1097/WAD.0b013e31815d19fe.

[9] W. J. M. Levelt, A. Roelofs, and A. S. Meyer, "A theory of lexical access in speech production," *Behavioral and Brain Sciences*, vol. 22, no. 01, Feb. 1999. DOI: 10.1017/S0140525X99001776.

[10] F. H. Guenther, *Neural control of speech.* Mit Press, 2016, ISBN: 9780262034715.

[11] F. L. Darley, *Motor speech disorders / Frederic L. Darley, Arnold E. Aronson, Joe R. Brown.* eng. Philadelphia: W. B. Saunders, 1975, ISBN: 9780721628783.

[12] J. Ogar, H. Slama, *et al.*, "Apraxia of Speech: An overview," *Neurocase*, vol. 11, no. 6, pp. 427–432, Dec. 2005. DOI: 10.1080/13554790500263529.

[13] S. Narayanan, K. Nayak, *et al.*, "An approach to real-time magnetic resonance imaging for speech production," *The Journal of the Acoustical Society of America*, vol. 115, no. 4, pp. 1771–1776, Apr. 2004. DOI: 10.1121/1.1652588.

[14] E. Bresch, Yoon-Chul Kim, *et al.*, "Seeing speech: Capturing vocal tract shaping using real-time magnetic resonance imaging [Exploratory DSP]," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 123–132, May 2008. DOI: 10.1109/MSP.2008.918034.

[15] M. Uecker, S. Zhang, *et al.*, "Real-time MRI at a resolution of 20 ms," *NMR in Biomedicine*, vol. 23, no. 8, pp. 986–994, Oct. 2010. DOI: 10.1002/nbm.1585.

[16] M. Fu, B. Zhao, *et al.*, "High-resolution dynamic speech imaging with joint low-rank and sparsity constraints," *Magnetic Resonance in Medicine*, vol. 73, no. 5, pp. 1820–1832, May 2015. DOI: 10.1002/mrm.25302.

[17] S. G. Lingala, B. P. Sutton, *et al.*, "Recommendations for real-time speech MRI," *Journal of Magnetic Resonance Imaging*, vol. 43, no. 1, pp. 28–44, Jan. 2016. DOI: 10.1002/jmri.24997.

[18] P. W. Iltis, J. Frahm, *et al.*, "High-speed real-time magnetic resonance imaging of fast tongue movements in elite horn players.," *Quantitative imaging in medicine and surgery*, vol. 5, no. 3, pp. 374–81, Jun. 2015. DOI: 10.3978/j.issn.2223-4292.2015.03.02.

[19] M. Fu, B. Zhao, *et al.*, "High-resolution dynamic speech imaging with joint low-rank and sparsity constraints," *Magnetic Resonance in Medicine*, vol. 73, no. 5, pp. 1820–1832, May 2015. DOI: 10.1002/mrm.25302.

[20] M. Fu, M. S. Barlaz, *et al.*, "High-frame-rate full-vocal-tract 3D dynamic speech imaging," *Magnetic Resonance in Medicine*, vol. 77, no. 4, pp. 1619–1629, Apr. 2017. DOI: 10.1002/mrm.26248.

[21] V. Ramanarayanan, S. Tilsen, *et al.*, “Analysis of speech production real-time MRI,” *Computer Speech & Language*, vol. 52, pp. 1–22, Nov. 2018. DOI: `10.1016/j.csl.2018.04.002`.

[22] J. Ogar, S. Willock, *et al.*, “Clinical and anatomical correlates of apraxia of speech,” *Brain and Language*, vol. 97, no. 3, pp. 343–350, Jun. 2006. DOI: `10.1016/j.bandl.2006.01.008`.

[23] A. Ozerov, E. Vincent, and F. Bimbot, “A General Flexible Framework for the Handling of Prior Information in Audio Source Separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1118–1133, May 2012. DOI: `10.1109/TASL.2011.2172425`.

[24] M. Burdumy, L. Traser, *et al.*, “Acceleration of MRI of the vocal tract provides additional insight into articulator modifications,” *Journal of Magnetic Resonance Imaging*, vol. 42, no. 4, pp. 925–935, Oct. 2015. DOI: `10.1002/jmri.24857`.

[25] X. Liu, L. Song, *et al.*, “A Review of Deep-Learning-Based Medical Image Segmentation Methods,” *Sustainability*, vol. 13, no. 3, p. 1224, Jan. 2021. DOI: `10.3390/su13031224`.

[26] N. O’Mahony, S. Campbell, *et al.*, “Deep Learning vs. Traditional Computer Vision,” in 2020, pp. 128–144. DOI: `10.1007/978-3-030-17795-9_10`.

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. Adaptive computation and machine learning. MIT Press, 2016, ISBN: 9780262035613.

[28] W. Di, A. Bhardwaj, and J. Wei, *Deep Learning Essentials: Your Hands-on Guide to the Fundamentals of Deep Learning and Neural Network Modeling*. Packt Publishing, 2018, ISBN: 1785880365.

[29] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in 2015, pp. 234–241. DOI: `10.1007/978-3-319-24574-4_28`.

[30] S. Erattakulangara and S. G. Lingala, “Airway segmentation in speech MRI using the U-net architecture,” in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, IEEE, Apr. 2020, pp. 1887–1890, ISBN: 978-1-5386-9330-8. DOI: `10.1109/ISBI45749.2020.9098536`.

[31] M. Ruthven, M. E. Miquel, and A. P. King, “Deep-learning-based segmentation of the vocal tract and articulators in real-time magnetic resonance images of speech,”

Computer Methods and Programs in Biomedicine, vol. 198, p. 105 814, Jan. 2021. DOI: 10.1016/j.cmpb.2020.105814.

[32] A. A. Taha and A. Hanbury, "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, no. 1, p. 29, Dec. 2015. DOI: 10.1186/s12880-015-0068-x.

[33] A. A. Taha and A. Hanbury, "An Efficient Algorithm for Calculating the Exact Hausdorff Distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2153–2163, Nov. 2015. DOI: 10.1109/TPAMI.2015.2408351.

[34] P. Virtanen, R. Gommers, *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020. DOI: 10.1038/s41592-019-0686-2.

[35] X. Li, X. Sun, *et al.*, "Dice Loss for Data-imbalanced NLP Tasks," Nov. 2019. DOI: 10.48550/arXiv.1911.02855.

[36] J. Ma, J. Chen, *et al.*, "Loss odyssey in medical image segmentation," *Medical Image Analysis*, vol. 71, p. 102 035, Jul. 2021. DOI: 10.1016/j.media.2021.102035.

[37] B. Maas, E. Zabeh, and S. Arabshahi, "QuickTumorNet: Fast Automatic Multi-Class Segmentation of Brain Tumors," in *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*, IEEE, May 2021, pp. 81–85, ISBN: 978-1-7281-4337-8. DOI: 10.1109/NER49283.2021.9441286.

[38] S. Firuzinia, S. M. Afzali, *et al.*, "A robust deep learning-based multiclass segmentation method for analyzing human metaphase II oocyte images," *Computer Methods and Programs in Biomedicine*, vol. 201, p. 105 946, Apr. 2021. DOI: 10.1016/j.cmpb.2021.105946.

[39] O. Oktay, J. Schlemper, *et al.*, "Attention U-Net: Learning Where to Look for the Pancreas," Apr. 2018. DOI: 10.48550/arXiv.1804.03999.

[40] P. Cerveri, M. Rossi, *et al.*, "CEL-Unet: distance weighted maps and multi-scale pyramidal edge extraction for accurate osteoarthritic bone segmentation in CT scans," 2022. DOI: 10.3389/frsip.2022.857313.

A | Appendix A

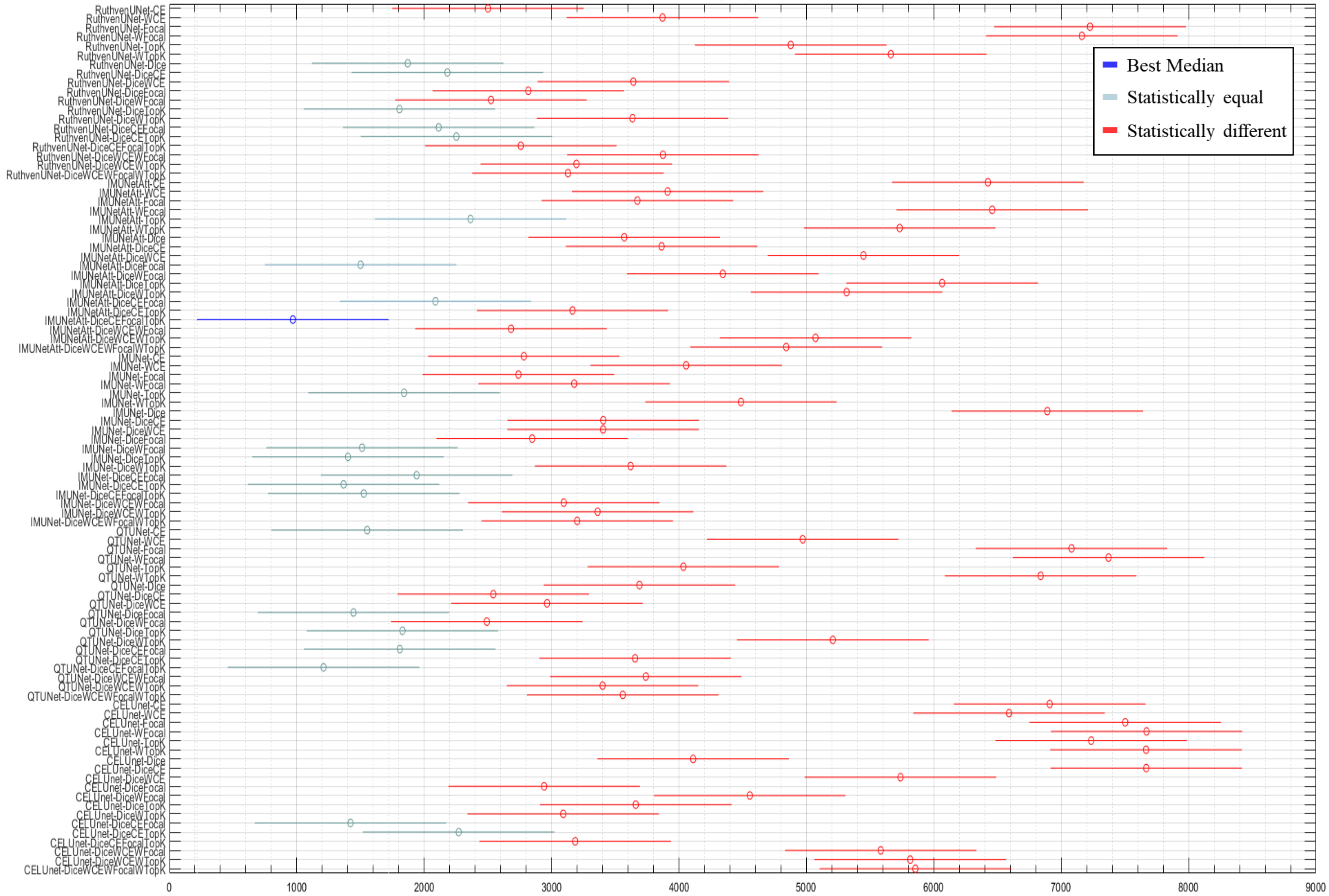


Figure A.1: Post Hoc test. 73 groups have mean ranks significantly different from IMUNetAtt-DiceCEFocalTopK

List of Figures

List of Tables

List of Symbols

Acronym	Description
nfvPPA	non-fluent/agrammatic variant primary progressive aphasia
CBS	corticobasal syndrome
PSP	progressive supranuclear palsy
ALS	amyotrophic lateral sclerosis
FTLD	frontotemporal lobar degeneration
AOS	apraxia of speech
SLP	Speech-Language Pathologists
dsMRI	dynamic speech MRI
ROI	Region of interest
XRMB	X-ray microbeam
EPG	electropalatography
EMA	electromagnetic articulography
UL	Upper Lip
HP	Hard Palate
SP	Soft Palate
TO	Tongue and Epiglottis
LL	Lower Lip and Jaw
HE	Head
BK	Background
HD	Hausdorff Distance
DICE	Dice coefficient
GCE	Global Consistency Error
OM	Overall metric
IQR	interquartile range
ePB	encoding processing blocks
dPB	decoding processing blocks
VTS-tool	Vocal Tract Segmentation tool

Acknowledgements

This thesis was carried out at Department of Electronics, Information and Bioengineering of Politecnico di Milano, in collaboration with Department of Neurology and Department of Radiology of University California San Francisco.

We would like to thank Professor Pietro Cerveri and PhD Maria Luisa Mandelli for giving us the opportunity to learn more about the field of deep learning and its countless clinical applications. We also thank them for giving us the opportunity to work in a stimulating multidisciplinary team.

Particularly, we would like to thank MSc Matteo Rossi for his constant willingness to support us and provide us with useful technical advice for achieving the objectives of this thesis.

We would like also to thank MD Shaun Ivan Muzic and PhD Eduardo Europa for the help and supervision provided during the manual segmentation phase.