POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Neuroengineering project: Segmentation of the vocal tract

Authors: **Benedetto Francesco,**
Bombacigno Angelica,
Fabiano Livia,
Pivetta Federico,
Santini Letizia

Prof: Prof. Cerveri Pietro
Tutor: Matteo Cavicchioli
Academic Year: 2023-24

Contents

1 | Context

1.1. Clinical overview

Magnetic resonance is a medical imagine technique, and due to the fact that it is non-invasive, it can be used to visualize human features in order to extract information. In this case, there were extracted information from the vocal tract, using *segmentation*, to perform some quantitative analysis.[1] This visualization provides information about shape, size, motion and position of the vocal tract and articulation; in particular, the segmentation considers the following part:

1. Background and vocal tract
2. Upper lip
3. Hard palate
4. Soft palate
5. Tongue
6. Lower lip
7. Head

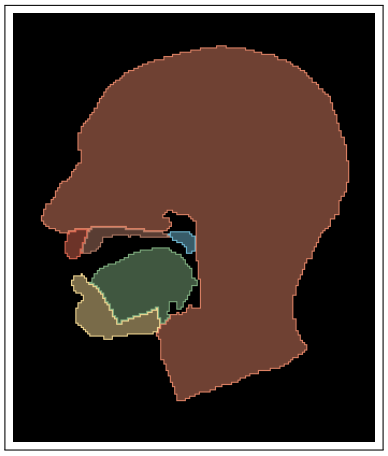


Fig.1.1 Segmentation of vocal tract visualized through the software Slicer.

1.2. Network

In order to segment in an efficient way, it can be used a deep learning algorithm; in particular, this project exploits the use of the IMproved U-net architecture (**IMU-net**).

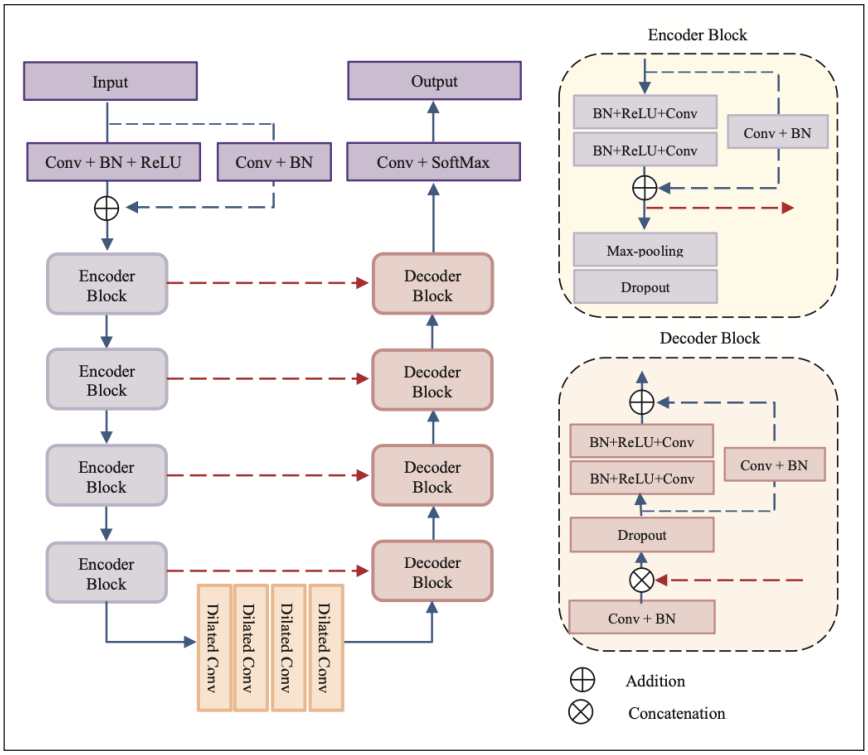


Fig.1.2 This is the network implemented in the project seen in detail.

1.3. Aim of the project

The aim of the project is to develop a neural network able to divide the vocal tract in the 7 segment listed above Sec.1.1. The workflow followed in this project is the following one:

1. *Given the fact that there is Gaussian noise embedded in the dataset, the imagines are pre-processed with a Gaussian filter Sec.2.2. Then, to improve the capability of correctly segment even the patient moving during the acquisition, it is performed a data augmentation Sec.2.3. Finally, the data are divided in order to fullfill the following requirements:*
    - 70% *s00001 (280 samples) and s00005 (150 samples) are used for the* ***training***

- 20% *s00002 (240 samples) is used for the* ***validation***
- 10% *s00005 (150 samples) is used for the* ***test***

2. *The IMU-net is implemented in the Chapt.3, following the features listed in [2] and modifying the parameters in order to assess the best model*
3. *The plot trend are computed in order to decide which is the best model to use. Hence, it is performed the final test of the model. Chapt.4*
4. *In the last section, there are the conclusion of the work. Chapt.5*

2 | Data preparation

In this subchapter, the methodologies adopted to load and process data have been described. In particular, the pre-defined schema has not been used. An independent and controlled method has been preferred.

2.1. Data division

The dataset is composed of four subject:

- s00001
- s00002
- s00004
- s00005

which contribute to the dataset with respectively 280, 240, 150, and 150 images. Therefore, different approaches have been used to divide data into training, validation and test sets. In order to have the same starting baseline, 4 directories - one for each patient - have been defined. Then data have been processed as defined in Sect.2.2 and augmented as defined in Sect.2.3.

The main approach to split dataset into training-validation-test sets has been through patient stratification. The subdividing percentages have been fixed to 0.7, 0.2 and 0.1 respectively. Patient stratification technique has been chosen as the main methodology because it minimizes sampling bias because the images from all the patients are equally distributed into training, validation and test sets.

Another approach was patient-specific cross evaluation. Among the 4 groups of patients, two of them were selected for training, one for validation and the remaining as a test-independent one. With the purpose of having minimum bias, different combinations of the patient have been made. Due to the huge computational effort required for each training, only 6 of them have been taken into consideration. The combinations have been

randomly taken from those having at least s00001 or s00002 in the training set, otherwise, the training set was not large enough. Finally, performances on the test sets have been averaged.

2.2. Pre-processing

Since images were affected by a gaussian noise, following procedures have been played out:

- **Gamma transformation** using $\gamma = 1.5$ in order to amplify the grey levels on the darker part of the spectrum;
- **Gaussian filter** with $\sigma = 0.4$ to eliminate noisy pixels;
- **Saturation** to black pixels that are below a predefined threshold in order to remove dots from the background.

This methodology has been confirmed by looking to the resulting images, to the signal to noise ration, **SNR**, and to the histogram of the difference of the original and filtered images. The latter one was bell-shaped suggesting that Gaussian noise has been removed. The comparison between the original image and the filtered one is depicted in fig.2.1

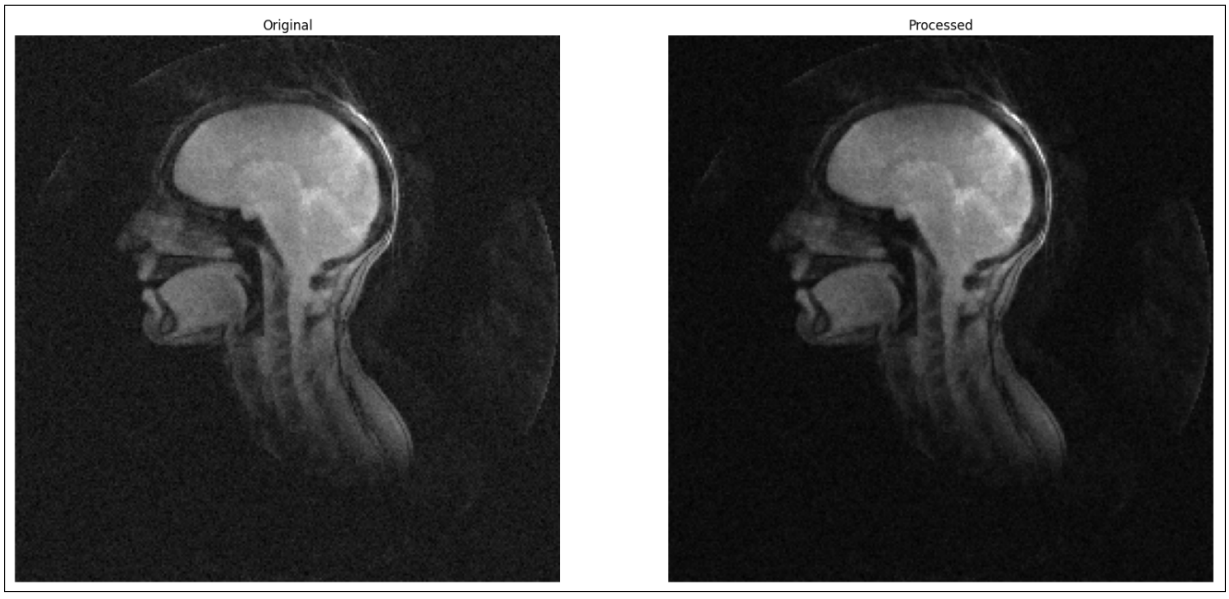


Fig.2.1 On the left, there is the imagine with the Gaussian noise; on the right, there is the imagine after the pre-processing.

After that, the imagines are saved in a vector.

2.3. Data augmentation

A data augmentation was performed using *Image Data Generator*, in order to increase the overall amount of data and to have full control of them as far as rotation and translation of patients' images are concerned; the augmentation was applied coherently on both X and y.

Having observed that the online procedure, which consisted in adding a layer on the model, did not give back the desired control on the label coordinate, data augmentation has been performed offline for each image and label pair. Due to memory limitation, it has been performed separately. Augmentation has been performed by applying randomly one or more of the following transformations:

- 10° of rotation
- 5 side translation
- 5 height translation
- 0.1% zoom

Data augmentation was applied only to training dataset, with *filling model nearest*. In this phase, augmentation was quite limited because the original data were homogeneous, therefore a strong augmentation would have compromised model's performance.

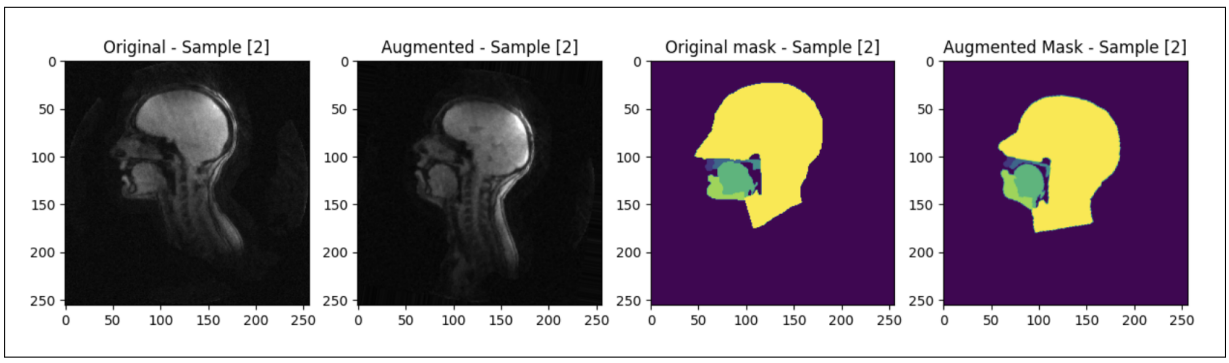


Fig.2.2 Data augmentation with a negative rotation of the subject.

Moreover, data augmentation has been done in order to increase the generalizing capability of the network. It has been performed in 500 randomly sampled training image/label pairs due to memory limitation. Then, a certain number of them has been concatenated to the training set. This number has been optimised through fine-tuning. The best model was obtained by concatenating 300 of them. Subsequently, results of the have been combined together and the average over single models have been taken , in order to have a

predicted model's measurement. However, in this specific case, augmentation was not undergone due to lack of memory and time available. As a final remark, results are much lower compared to the ones belonging to the classical model, probably due to the far reduced amount of samples.

3 | Workflow

3.1. Network implemented

IMproved U-Net (IMU-Net) [2] has been implemented as the main model. It is based on U-Net architecture which is the architecture reference in image segmentation. The encoding part is able to extract global - due to a large receptive field - features of the whole image, while the decoding part, which receives input from the bottleneck and skip connections, guarantees to propagate information about the position in the image. Different from traditional U-Net, encoding and decoding blocks incorporate residuals.

More in detail IMU-Net, represented in fig.3.1, is composed by one convolutional block (32 filters and kernel=(3,3)) followed by a batch normalization and ReLU activation layers and added to a convolutional layer (32 filters and kernel=(1,1)) followed by a batch normalization layer.

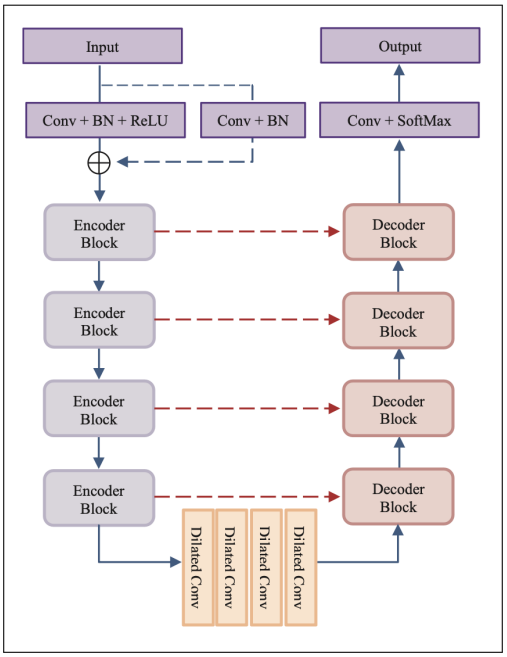


Fig.3.1 This is the model use in the project, in particular the IMU-Net.

Then, like in U-Net architecture, there are 4 encoding blocks each one resulting in one output layer which enters in the following layer and one skip layer which is used as skip connection.

The internal structure of each encoding block is represented in fig.3.2. In [2], no information about the number of filters and kernel size of each convolutional layer in the contraction blocks were provided, therefore those implemented in U-Net architecture were used. In particular, kernel size has been fixed to (3,3) for each convolutional layer while the number of filters double block by block. The last 2 layers of each encoding block are respectively a max-pooling layer (size=2 and stride=2) which allows to half each spatial dimension and a dropout layer (rate=0.5) which allows overfitting to prevented.

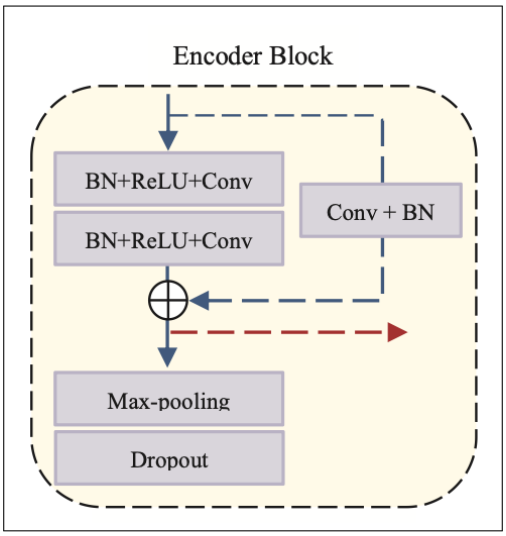


Fig.3.2 Detail on encoding of the IMU-Net.

The output of the fourth encoder block enters the bottleneck section which is composed of 4 dilated convolution layers each one characterized by an equal number of filters (1024) and the same kernel size (3,3) but with an increasing dilation rate, respectively 1, 2, 4, 8. This last hyperparameter is fundamental to increasing the receptive field.

For what concerns the decoding section, it is composed of 4 decoding blocks. Each decoder block, reported in fig.3.3, takes as input the previous layer and the corresponding skip layer. The previous layer enters a first layer called transposed convolution (half filter size of the input layer, kernel=(1,1) and strides=2) that is fundamental to double the spatial dimensions of the activation map. This layer is followed by a batch normalization layer.

Then this first output is concatenated to the respective skip layer in order to combine global information (from the bottleneck layer) and spatial information (from the skip

connection). The resulting layer enters a sequence of dropout, convolution, batch normalization and ReLU activation using a residual connection as represented in fig.3.3. For what concerns the hyperparameters, those implemented in U-Net were followed because no information in [2] was provided. Therefore, each decoder block ends with a layer that has double each spatial dimension and half the number of activation maps in respect to the input layer or, equivalently has the same dimension and same number of activation maps of the entering skip layer.

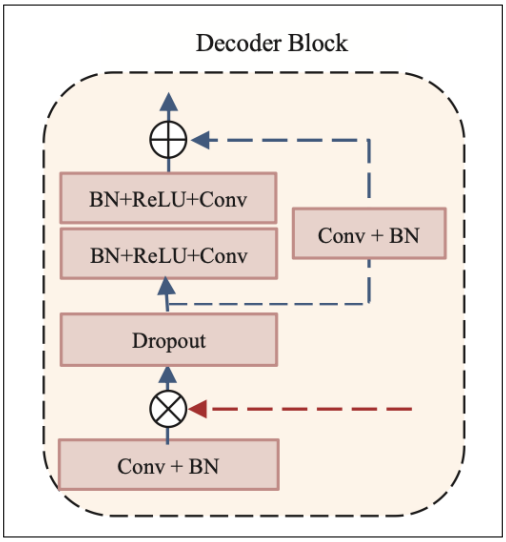


Fig.3.3 Detail on decoding of the IMU-Net.

The output layer of the fourth decoder block enters the last convolutional layer that is characterized by a kernel size equal to (1,1) and a number of filters equal to the number of classes to segment (7). Softmax activation function has been used in order to have an output that can be interpreted as a probability (for each pixel, the value in the corresponding X activation map represent the probability of being classified by IMU-Net as belonging to class X). The network implemented is shown more in detail in fig.3.4, it has been obtained using *layered_view* tool from visual keras. Since, it is not able to figure skip connections, they have been added using another software.

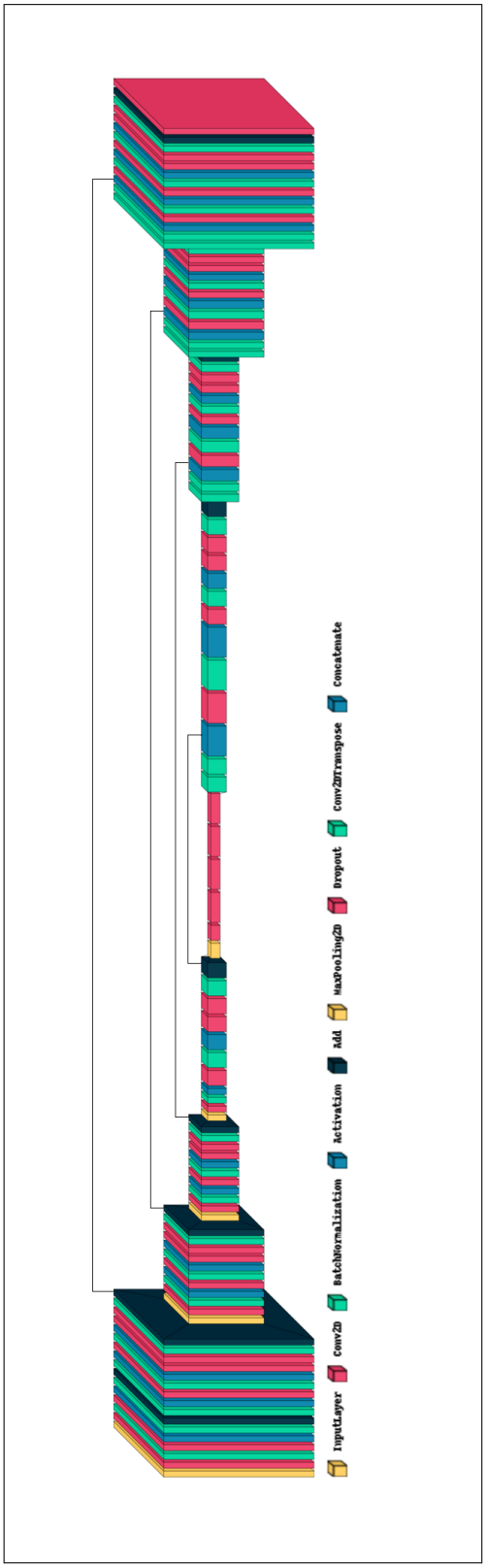


Fig.3.4 This is the network implemented in the project.

3.2. Training methodology

In the training phase, we monitored different metrics. In particular, **DICE coefficient**[3], **accuracy** and **precision** have been used to monitor the training, while **recall** and **mean_IoU** [4] (*IoU*, *Intersection over Union*) were also used for the evaluation of the final model. For what concerns the loss function, we implemented a weighted cross entropy. It was essential to increase the importance of the under-represented classes (e.g. upper lip) with respect to numerous classes (e.g. background).

Adam optimizer has been chosen as the loss optimizer, becuase it is the most common used one. To control and visualize the model behaviour epoch by epoch, **VizCallback** has been implemented. It allowed us to visualize and save the resulting segmentation of a fixed image from the validation set compared to the ground truth. To prevent overfitting two other callbacks have been implemented: early stopping and ReduceLROnPlateau (learning rate scheduler).

The proposed model has been trained using the following parameters:

- batch_size = 8
- epochs = 50
- learning_rate = 0.001
- early_stopping_monitor_metric = 'val_Mean_DICE'
- early_stopping_patience = 10
- lr_scheduler_monitor_metric = 'val_loss'
- lr_scheduler_patience = 4
- lr_scheduler_factor = 0.5
- lr_scheduler_min = $1e-5$

4 | Test and validation

In this chapter's section, the following subdivision of the datasets will be considered as the starting point of the discussed analysis:

- 70% of each subject's dataset is used for training
- 20% of each subject's dataset is used for validation
- 10% of each subject's dataset is used for test

	s0001	s0002	s0004	s0005	Total
Training	196	168	105	105	574
Validation	56	48	30	30	164
Test	28	24	15	15	82

Table 4.1: Dataset division for each part of the work

This subdivision is important to be specified because, in the last part of analysis, the above mentioned dataset has been modified in accordance to an observation that will be discussed later.

4.1. Validation

The metrics used during both the phase of training and validation are the following:

1. **Weighted cross entropy loss function**: in the initial part of the analysis a "*cross entropy loss function*" was used, but then the following analysis was conducted using a weighted one. This choice was taken as the results showed some underrepresented classes so the aim was to increase the impact of these ones.
2. **Accuracy**
3. **Precision**
4. **Dice coefficient**

The aim of analyzing the previous metrics both in the training and in the validation phases was not only to monitor the overfitting but also to study which model was best able to generalize the features in the validation dataset. In particular, this idea was taken into account to decide which was the model that returned the best results.

4.2. Trials

Different types of analysis have been performed. In particular, the focus was on changing different types of parameters during the training in order to study which model returned the best results. The following diagram shows the workflow that has been followed and the whole sets of models trained and validated. In particular, the starting point was the dataset mentioned in the previous section; then, as a first step, some changes in the characteristics of the dataset were applied:

- "*Non-filtered, non-augmented*": refers to a training set with images that were not filtered and to which the augmented images were not added. This kind of analysis had the purpose of evaluating how much the network was able to perform with the original images. Moreover, this model represents the baseline that must be overcome.
- "*Filtered, non-augmented*": refers to a training set with images that are pre-processed but to which augmented images were not added;
- "*Filtered, augmented*": refers to a training set with filtered images to which the augmented images were added. Some trials with different numbers of augmented images have been performed. In particular, starting from an original training dataset made of 574 images, 500, 300 and 200 augmented images were added. Finally, without providing details, the model with 300 augmented images was demonstrated to be the best-performing trial.
- "*Filtered, new augmentation*": refers to a training set with filtered images and with a different amount of filtered images and augmented images in the training dataset. In particular, it was made of:
  - 374 filtered images
  - 200 augmented images

This analysis aimed to evaluate how much the augmented images could impact the results when their amount was significantly related to the normal filtered ones.

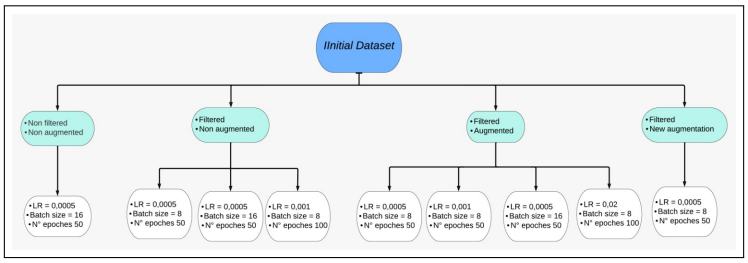


Fig.4.1 These are the combinations of parameters develped in the project.

Then, under each different training dataset, many different analyses were performed; the main parameters that were taken into account were:

- Learning Rate
- Batch size
- Number of epochs

The metrics monitored in the callbacks, their patiences and the factor of the learning rate scheduler were kept fixed. '**val_mean_DICE**' has been selected as the monitoring metric in the callbacks because it is a good estimator for both over-estimation and under-estimation with respect to accuracy and precision. **Patiences** have been fixed respectively to 10 and 4 for early_stopping and the scheduler, while the scheduler factor has been fixed to 0.5 to prevent overfitting.

4.2.1. Results

The results provided by each model can be summarized in the following table. In the left column, the model evaluated is identified through an initial identification number (1, 2, 3 or 4):

- 1 → *Non-filtered, non-augmented*
- 2 → *Filtered, non-augmented*
- 3 → *Filtered, augmented*
- 4 → *Filtered, new augmentation*

Then, under each group of training dataset, the number of the corresponding trial is identified through the '.number' For instance, "2.3" identifies the third trial in the second group that is to say the trial under the "Filtered, non-augmented" group with the following characteristics: LR = 0.001, Batch size = 8 and Number of eras = 100

	Training				Validation			
	Precision	**DICE**	**Accuracy**	**Loss**	**Precision**	**DICE**	**Accuracy**	**Loss**
1	**0.9845**	**0.8468**	**0.9843**	**0.0497**	**0.9856**	**0.8482**	**0.9855**	**0.0737**
2.1	**0.9847**	**0.8576**	**0.9845**	**0.0463**	**0.9819**	**0.8533**	**0.9817**	**0.0930**
2.2	**0.9836**	**0.8394**	**0.9833**	**0.0511**	**0.9870**	**0.8522**	**0.9868**	**0.0769**
2.3	**0.9835**	**0.8453**	**0.9833**	**0.0497**	**0.9870**	**0.8648**	**0.9868**	**0.0830**
3.1	**0.9855**	**0.8839**	**0.9853**	**0.0401**	**0.9848**	**0.8637**	**0.9877**	**0.1123**
3.2	**0.9852**	**0.8340**	**0.9850**	**0.0441**	**0.9885**	**0.8787**	**0.9887**	**0.0988**
3.3	**0.9834**	**0.8241**	**0.9831**	**0.0508**	**0.9811**	**0.8173**	**0.9808**	**0.0927**
3.4	**0.9818**	**0.8318**	**0.9815**	**0.0579**	**0.9840**	**0.8155**	**0.9738**	**0.1207**
4	**0.9504**	**0.6505**	**0.9492**	**0.2250**	**0.9447**	**0.6578**	**0.9441**	**0.1339**

Table 4.2: Results obtained by each model.

Best model

To make an appropriate evaluation of the results obtained, the weighted sum of 'precision', 'DICE', and 'accuracy' of the validation of each model was performed. This equation gave an idea of which model had the best performance.

$$Validationsum = \frac{Precision_{validation} + DICE_{validation} + Accuracy_{validation}}{3} \tag{4.1}$$

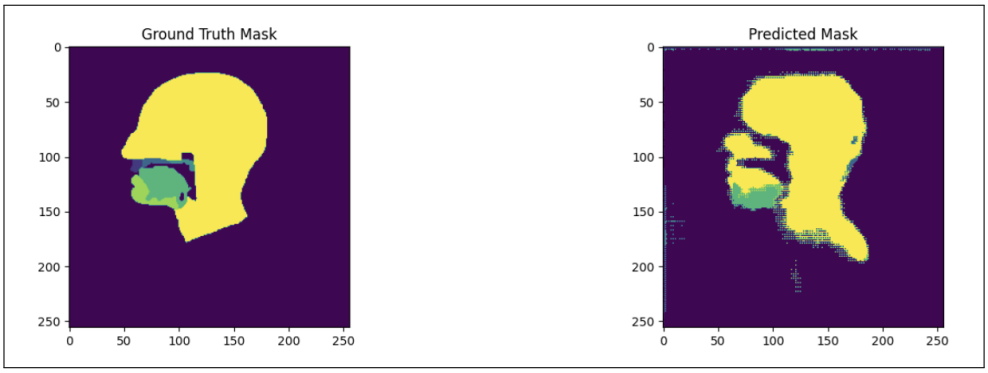
In particular, the model that returned the highest value of "Validation sum" was the 3.2

$$Validationsum_{3.2} = \frac{0.9885 + 0.8787 + 0.9887}{3} = 0.952 \tag{4.2}$$

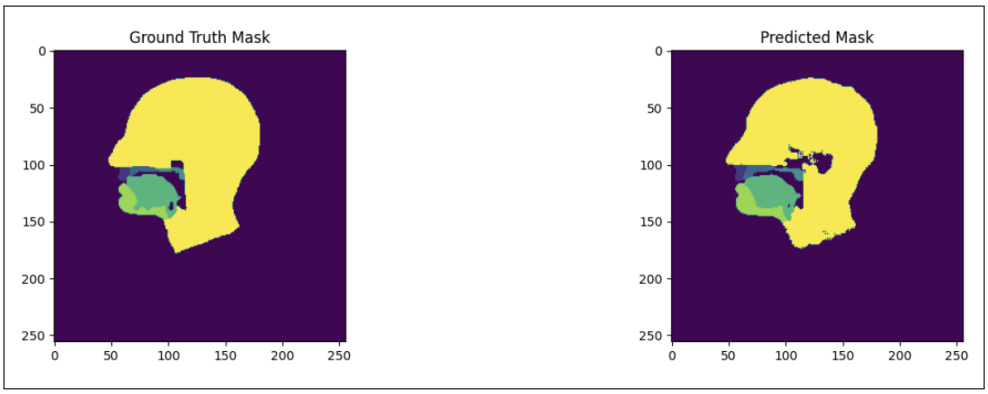
The model 3.2 had a "*Filtered, augmented*" training dataset with the following parameters:

- Learning Rate = 0.001
- Batch size = 8
- Number of eras = 50

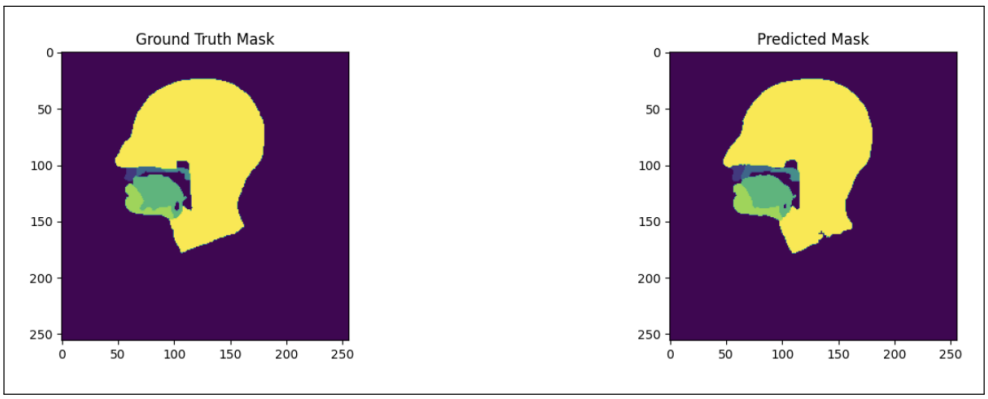
As discussed in the previous chapter, after the training, the predicted images (viz_callback function) and the graphs of the metrics both during the training and the validation with respect to the epochs were plotted. According to the best model obtained, the predicted images and the graphs are here reported. The images show the comparison between the '**ground truth mask**' and the '**predicted mask**' in the first, in the 20^{th} and in the last epochs.



(a) 1st eras

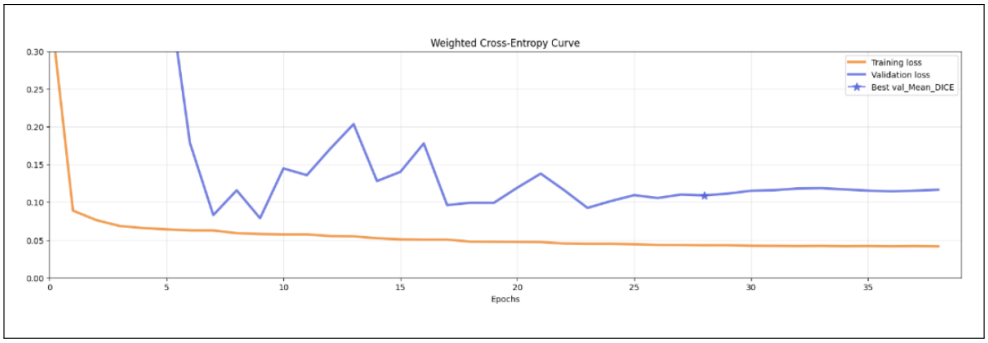


(b) 20th eras

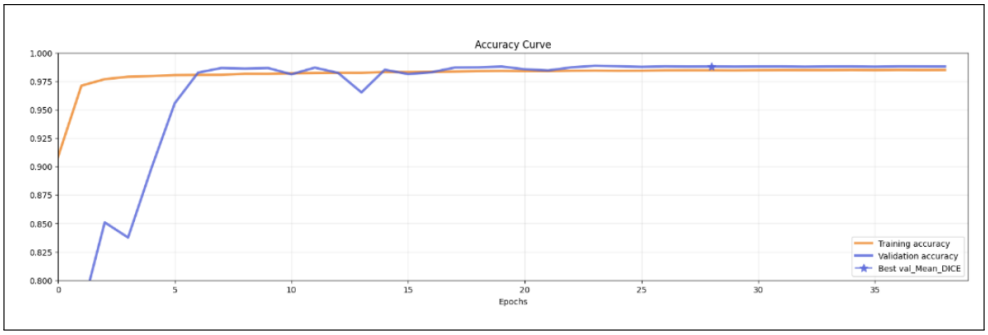


(c) 50th eras

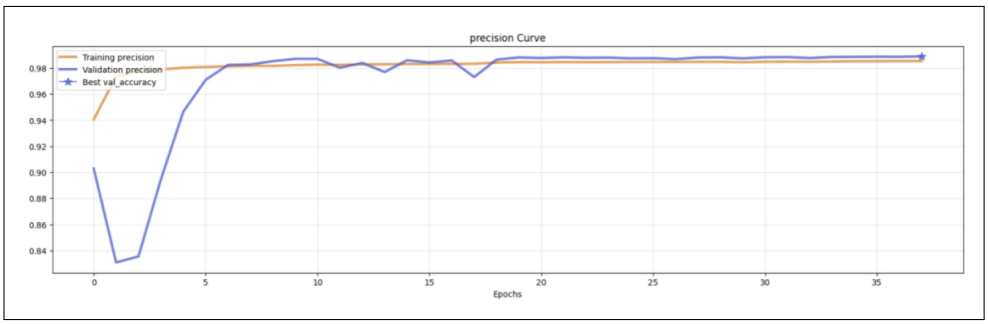
Figure 4.2: Comparison between the 'ground truth mask' and the 'predicted mask'



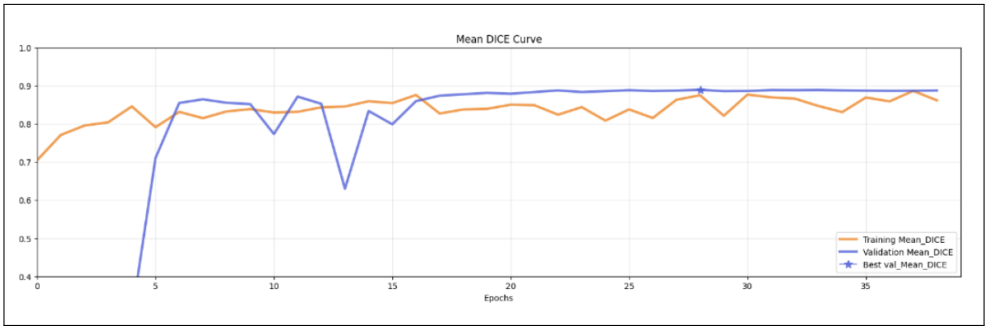
(a) Weighted cross entropy curve



(b) Accuracy curve



(c) Precision curve



(d) Mean DICE curve

Figure 4.3: Trend plot of the previous training

Other interesting results

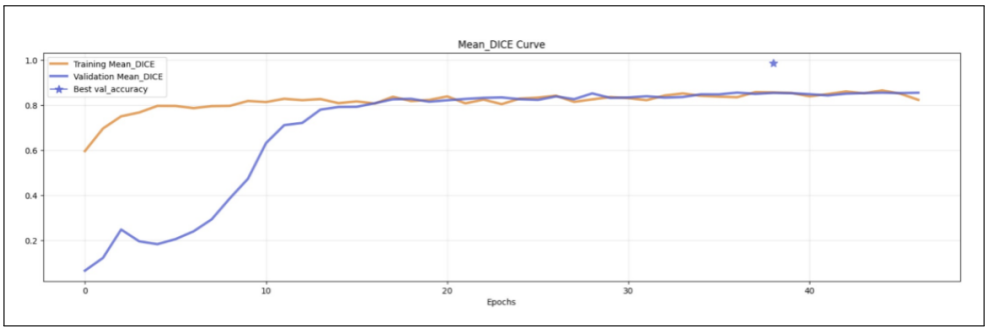
According to the results reported in the table, some other interesting considerations were made:

- The training of models having "number of epochs" = 100 stopped automatically around the 60^{th} epoch. This demonstrated that the network that was implemented performed and returned accurate results with a quite small number of epochs.
- The model under the "Filtered, new augmentation" consideration, returned bad results: the value of the loss function during the validation was too high demonstrating overfitting. As a matter of fact, it was concluded that a too high number of augmented images with respect to that for filtered images in the training dataset generates overfitting. The network is not infact able to generalize the features learned during the training set.
- The first model ("Non filtered, non augmented") showed very promising values of the metrics considered with respect to those obtained for processed images. This demonstrated that the network implemented was able not only to correctly process the images but also to work properly with the gaussian noise. However, it is important to notice that the amount of noise in the original dataset is not significant. The Mean-Dice curve and the predicted mask in the last epoch are here reported in fig.**??**

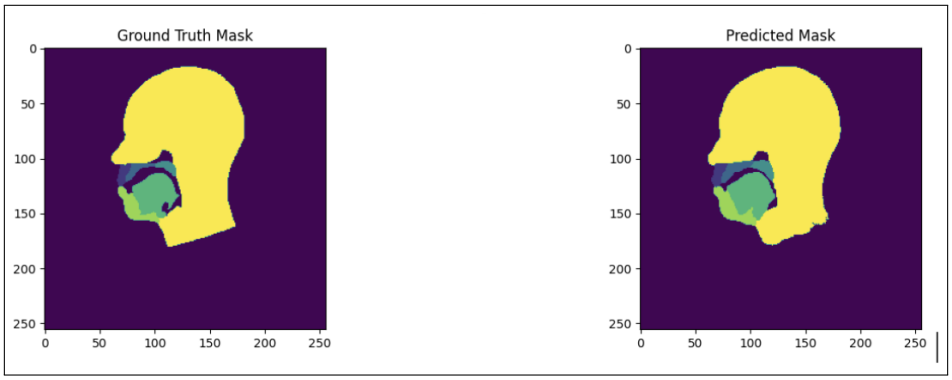
The model that returned the worst results is "Filtered, augmented" with parameters:

- Learning Rate = 0.02
- Batch size = 8
- Number of epochs = 50

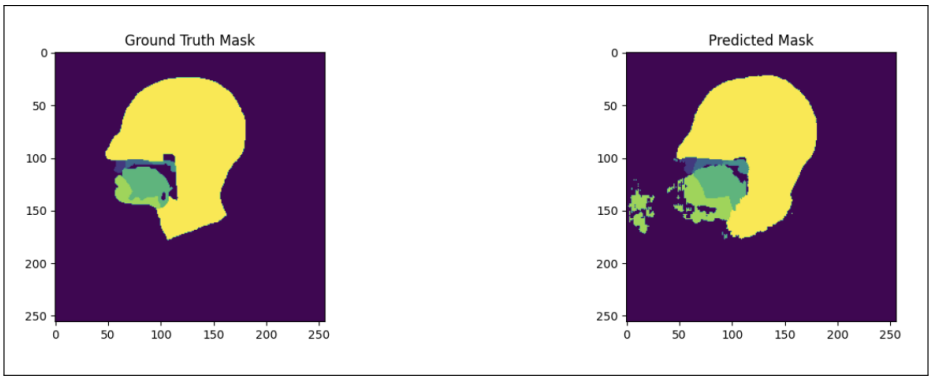
This was simply a proof of the fact that a too high value of the learning rate generates instability, divergence and overshooting that produces an excessive change in weights during each optimization step. The following figures represent the '**ground truth mask**' with respect to the '**predicted mask**' at the 45^{th} and 50^{th} epoch (see fig.4.4).



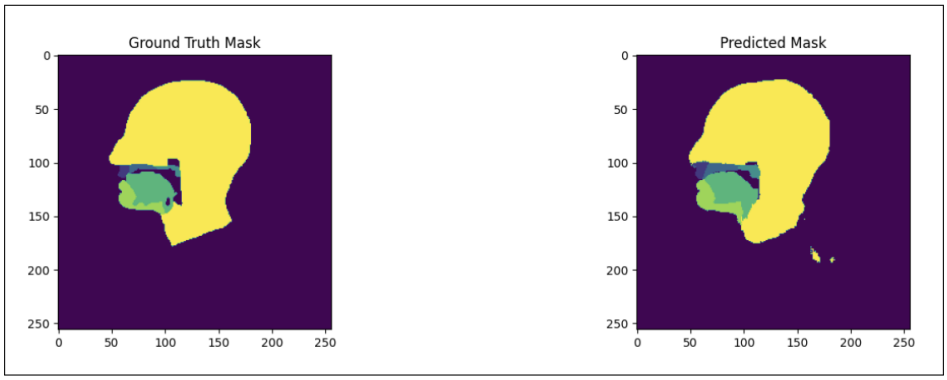
(a) Mean_DICE curve for the best model



(b) Difference between "ground truth mask" and "predicted mask" for the best model



(c) Difference between "ground truth mask" and "predicted mask" for the 45^{th} eras



(d) Difference between "ground truth mask" and "predicted mask" for the 50^{th} eras

Figure 4.4: Other interesting results.

4.3. Test

Once the training and the validation phase were correctly performed, the test phase was taken into account. In particular, as a post-process analysis, the argmax on the predicted image was performed in order to return the index of the class with the highest probability. Moreover, in this phase of the project other two parameters were considered with respect to those studied in the previous phases:

- **Mean IoU**
- **Recall**

Talking about the **Mean Iou** particular metric, some considerations must be done. Initially, this parameter was studied also during training and validation; however, the values obtained in the two phases were useless for the purposes of this project. In particular, the mean IoU was implemented through the Tensor Flow library but it resulted to be cryptically explained. As a matter of fact, even if the model demonstrated to correctly segment the images, the values obtained during the training were often too much different from those obtained in the validation. In the end, this parameter was evaluated only after the argmax as it returned more affordable values. The **Recall** parameter is crucial to correctly identify all positive classes correctly identified. Moreover, in order to have a global evaluation of the performances of the model, two analysis were performed:

1. The precision, the recall and the mean DICE were calculated for each class.
2. The confusion matrix was implemented. In particular, the values reported in the matrix were weighted along each column: the classes in the column represented the predicted classes while those in the row represented the true classes.

Before providing the results obtained for the best model described in the previous section, it is important to underline that the test set was used only once when the best model had been identified. The reason for this consideration was to properly simulate a realistic use of the model and not to try to obtain the best performances. If the parameters of the test set had been used to evaluate the performance of the model, overfitting would have occurred.

Dice_no_argmax	0.888
Dice_argmax	0.893
IoU	0.674
Accuracy	0.987
Precision	0.8393
Recall	0.9704

Table 4.3: The results of the test set obtained for the best model discussed above are shown

Class	Precision	Recall	Mean DICE
0: background and vocal tract	0.9938	0.9926	0.9948
1: upper limb	0.7128	0.9770	0.8145
2: hard palate	0.7115	0.9554	0.7961
3: soft palate	0.6367	0.9610	0.7744
4: tongue	0.9242	0.9720	0.9585
5: lower lip	0.9153	0.9681	0.9532
6: head	0.9807	0.9670	0.9823

Table 4.4: The results of the test set obtained for the best model discussed above are shown, with a detailed list of each class

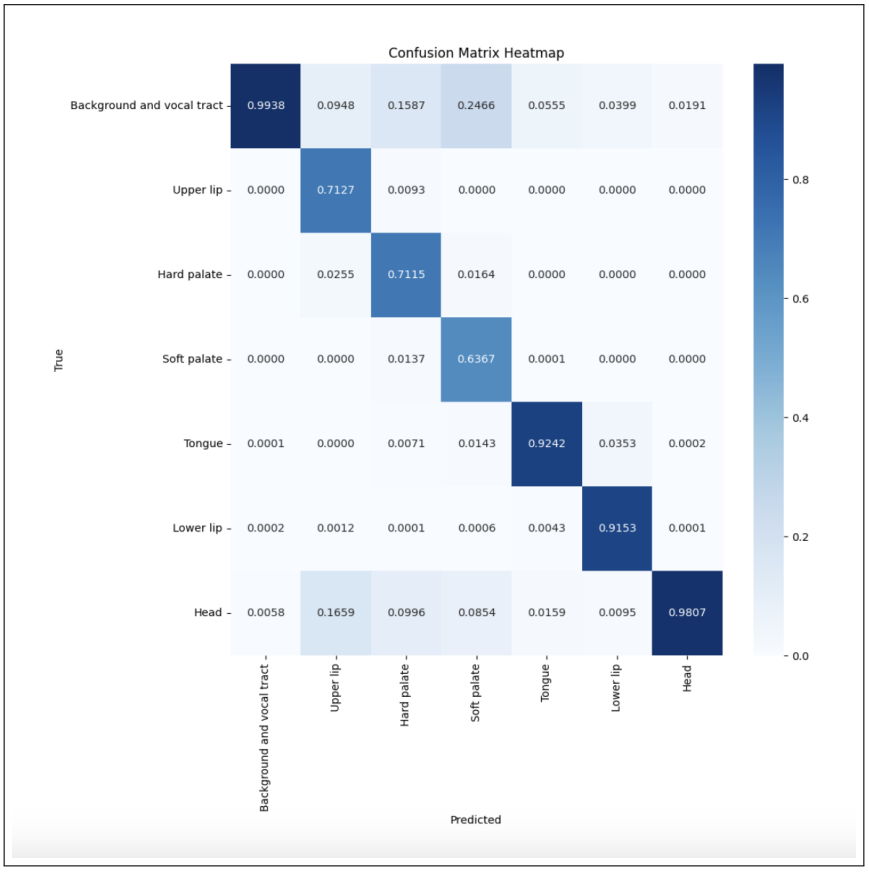


Fig.4.5 Confusion matrix for the best model.

As shown by the confusion matrix, the model properly classifies all the classes. The classes that presented the worst classification were the upper lip, hard palate and soft palate as they were misclassified respectively in 17% in head and 16 % and 25% in background and vocal tract. It is reasonable as they are the smallest parts in the anatomy of the subjects. Notice that the dice coefficient has been computed before and after the postprocessing in order to have an estimation of the confidence of the model. The two metrics are similar, therefore the model can be considered truthful. For what concerns precision and recall, they were computed as "macro". It means that precision and recall are computed class by class, and then, the overall result is equal to the average of them. This estimation has been preferred to the corresponding metrics computed pixel by pixel because it improves the effect of underrepresented classes that are the most difficult to classify.

4.4. Cross-evaluation

The final analysis was performed using a different approach of choosing the training, validation and test sets. In particular, all the possible combinations of subjects' images were used in the best model obtained before. Two different subjects' datasets were entirely used during training, while the other two subjects' datasets were used for validation and test. All the images were pre-proccesed while they were not augmented due to memory limitations.

First combination	s0001 and s0002 used for training
	s0004 used for validation
	s0005 used for test
Second combination	s0001 and s0004 used for training
	s0005 used for validation
	s0002 used for test
Third combination	s0001 and s0005 used for training
	s0004 used for validation
	s0002 used for test
Fourth combination	s0002 and s0004 used for training
	s0001 used for validation
	s0005 used for test
Fifth combination	s0002 and s0004 used for training
	s0005 used for validation
	s0001 used for test
Sixth combination	s0002 and s0005 used for training
	s0004 used for validation
	s0001 used for test

Table 4.5: Different combinations considered for the last analysis

Obviously, some of these analysis returned quite bad parameters because of the imbalance in the number of images in each dataset. For instance, in the sixth combination the dataset had:

- 390 images for the training dataset
- 150 images for the validation dataset
- 280 images for the test

In particular, reporting the confusion matrixes of the models obtained by the best and the worst combinations. The worst combination, as discussed above, is the 6th one.

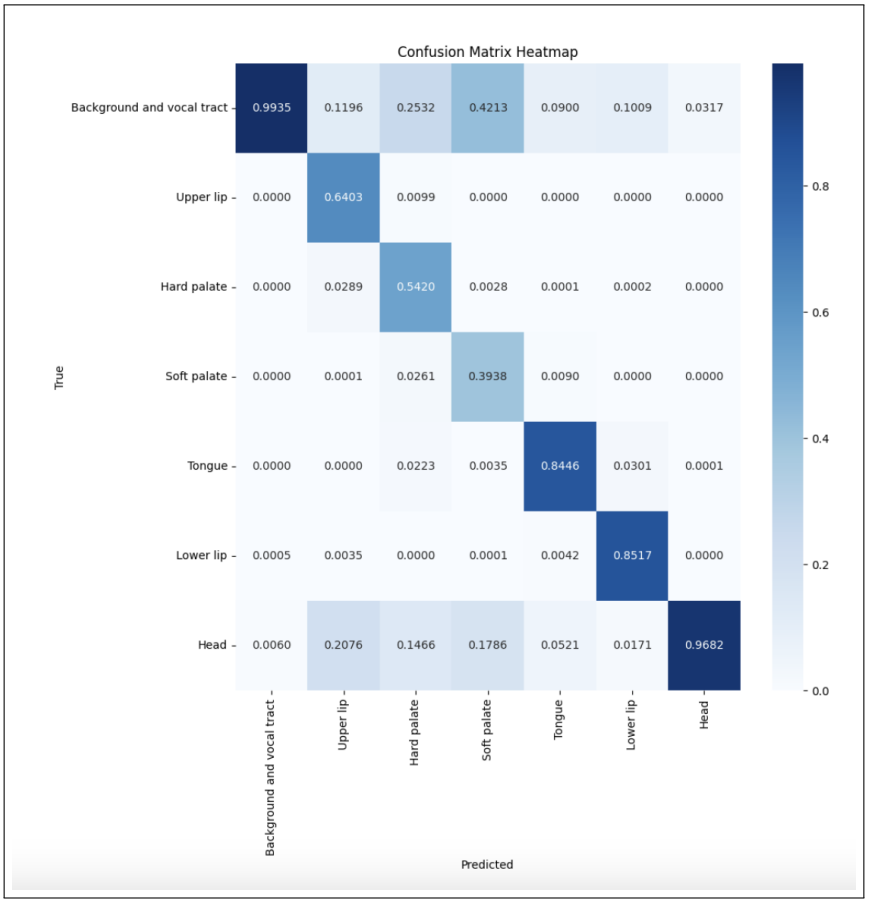


Fig.4.6 Confusion matrix for the worst combination, the 6th one.

It is clear from the confusion matrix that some classes, such as the soft palate and the hard palate are badly segmented. On the other hand, the combination that provided the best results was the 2nd one.

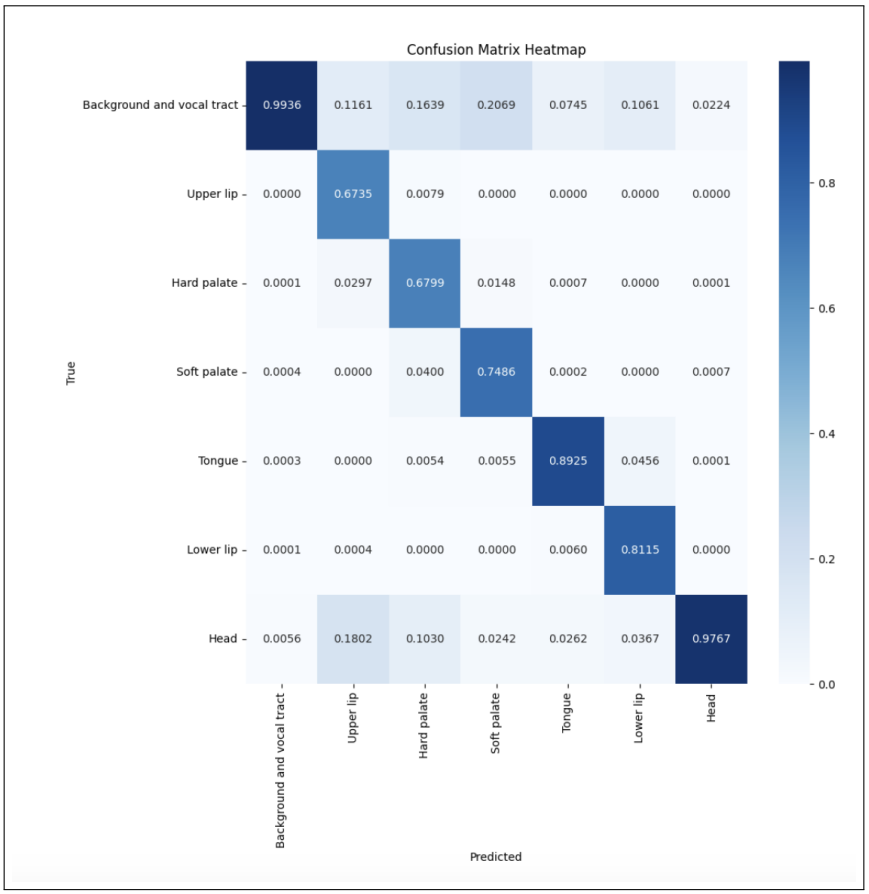


Fig.4.6 Confusion matrix for the best combination, the 2nd one.

Despite the soft palate resulted to be, also in this case, the worst segmented class, the performances of this combination of dataset demonstrated to be the best one.

Finally, the performances of all these models were combined in order to have a less biased evaluation of the model. This approach has been called **patient-divided-cross-evaluation**. All the test metrics have been averaged, and the results are the following:

Dice	0.8230
IoU	0.7569
Accuracy	0.9788
Precision	0.7743
Recall	0.9116

Table 4.6: The Result of the patient-divided-cross-evaluation discussed above are shown.

Class	Precision	Recall
0: background and vocal tract	0.9900	0.9890
1: upper limb	0.6065	0.9704
2: hard palate	0.6023	0.9132
3: soft palate	0.5254	0.7344
4: tongue	0.8836	0.9017
5: lower lip	0.8481	0.9264
6: head	0.9642	0.9463

Table 4.7: The Result of the patient-divided-cross-evaluation discussed above are shown, with a detailed list of each class.

This estimation of the model results less performing than the 3.2 model described before. This approach is probably less affected by sampling bias and it produces an estimation of the model that is more realistic because it uses an independent test set. On the other hand, having a small number of patients and images does not permit the model to learn as much as model 3.2.

4.5. Final analysis

In the end, a cross-validation analysis had been performed. Cross-validation is a technique used in machine learning to evaluate the performance of a statistical model and mitigate the risk of overfitting. Its main objective is to provide a more accurate estimate of a model's performance on data. In particular, four different combinations of dataset were considered: three out of four subjects' dataset were used for training, while the last one was used for validation:

1. $s0001, s0002, s0004$ training and $s0005$ validation $\rightarrow$ combination A
2. $s0001, s0002, s0005$ training and $s0004$ validation $\rightarrow$ combination B
3. $s0002, s0004, s0005$ training and $s0001$ validation $\rightarrow$ combination C
4. $s0001, s0004, s0005$ training and $s0002$ validation $\rightarrow$ combination D

The aim of this kind of analysis was to understand and give a numerical evaluation of the model's capability to generalize target features. Infact, changing the dataset used for training, had the purpose of avoiding overfitting and prevent the model from not becoming able to generalize features coming from other subjects' dataset.

Each combination of the dataset of training and validation was used in the best model described above and the parameters listed below were considered:

- DICE without argmax

- DICE with argmax
- accuracy
- Precision
- Recall
- IoU with argmax

Moreover, in order to have a more general estimate of the generalization capability of the model, the mean of each parameter in all the four different combinations was computed. For instance, according to the Accuracy, the mean was computed as following:

$$Mean_{Accuracy} = \frac{Accuracy_{combA} + Accuracy_{combB} + Accuracy_{combC} + Accuracy_{combD}}{4} \tag{4.3}$$

The results obtained are shown in the following table:

	Comb. A	Comb. B	Comb. C	Comb. D	Final mean
DICE without argmax	0.865	0.9635	0.848	0.8132	0.8723
DICE with argmax	0.8744	0.8698	0.8549	0.8249	0.8560
Accuracy	0.9809	0.9794	0.9836	0.9690	0.9782
Precision	0.8152	0.8145	0.7863	0.7774	0.7983
Recall	0.9617	0.9492	0.9637	0.9263	0.9502
IoU	0.7467	0.8474	0.6727	0.7819	0.7620

Table 4.8: The Result of the cross-validation discussed above are shown, with a detailed list of each combination.

The '*final Mean*' represents the mean of each parameter above all the four possible combination; as it presents high values for almost all the mean parameters, it is reasonable thinking that it will provide successfull results. However, as the study was performed just in term of training and validation, a quantitative analysis about the overfitting could not be done; infact, a test dataset coming from an external subject under analysis, would had been necessary in order to evaluate the capability of the model to overfit or underfit. In general, from a clinical point of view, the test dataset would come from a patient's MR images and of which the segmentation of the vocal tract is the clinical purpose.

5 | Conclusions

In conclusion, the project aimed to develop a neural network based on the IMproved U-net architecture for the segmentation of the vocal tract into seven distinct classes. The utilization of magnetic resonance imaging allowed for the extraction of valuable information regarding the shape, size, motion, and position of various vocal tract components. The segmentation process involved the use of a deep learning architecture, IMproved U-Net. This architecture was chosen based on its ability to efficiently segment images, incorporating features such as residual units and skip connections to enhance performance. The workflow of the project was carefully structured, starting with the preprocessing of images to address Gaussian noise and the implementation of data augmentation techniques to improve the model's ability to handle patient movement during image acquisition. The dataset, consisting of images from different subjects, was divided into training, validation, and test sets, ensuring a balanced representation of patients in each subset.

The training phase involved monitoring various metrics, including DICE coefficient, accuracy, precision, recall, and mean Intersection over Union (IoU). These metrics provided insights into the model's performance during both training and validation stages. Several trials were conducted to optimize model parameters such as learning rate, batch size, and number of epochs. The results indicated that the model trained on filtered images with augmentation (*trial 3.2*) yielded the best overall performance.

Further research and refinement could enhance the model's generalization capabilities and address specific challenges in vocal tract segmentation, contributing to advancements in medical imaging technology. The achieved performance of the developed neural network, based on the IMproved U-Net architecture, is indeed satisfactory. However, for a more robust evaluation and real-world applicability, it is essential to test the model on a dataset sourced from diverse medical centers. This would ensure independence and generalizability, providing a broader spectrum of images for the training set.

While the preprocessing steps contributed to the model's success, further investigation is warranted to assess its performance on raw MRI images with various noise levels. Testing with different noise patterns will shed light on potential challenges and robustness

in real-world scenarios. Recognizing the inherent challenge in classes with fewer pixels, it is acknowledged that the model's performance on these classes may be comparatively lower. Despite this, the overall performance remains commendable, demonstrating the network's ability to handle diverse anatomical features.

To enhance data augmentation, it was proposed to incorporate mirroring of training images, considering variations in patient positioning during MRI scans. Introducing shifts and rotations could further diversify the dataset, accounting for potential variations in head orientation during scans, which may vary between left and right positions. With increased computational power, expedited training could facilitate more efficient fine-tuning, potentially leading to improved model performance. Additionally, exploring alternative neural network architectures, such as augmenting the IMU-net with additional layers or considering architectures from the literature, may yield insights into further optimization possibilities. In pursuit of continuous improvement, it is suggested to experiment with ensemble methods, specifically bagging or boosting, leveraging the models developed in this study. Employing a majority voting approach within an ensemble can help mitigate individual model weaknesses, enhancing overall segmentation robustness.

In conclusion, this project has achieved commendable results through the successful implementation and optimization of the IMproved U-net architecture for vocal tract segmentation. The systematic evaluation of various training scenarios has demonstrated the efficacy of the model in effectively handling the complexities associated with vocal tract imaging. The outcomes obtained are satisfactory, highlighting a significant accomplishment in this research.

Furthermore, while the current model has shown promising results, it was recognized the importance of ongoing research. The focus remains on testing the model on diverse datasets, exploring its resilience to noise, refining data augmentation strategies, and delving into advanced neural network architectures. These continued efforts aim to provide valuable insights into medical image segmentation, with the ultimate goal of advancing the applicability of this model in clinical settings.

5.1. Clinical context

This model has some clinical application and can be used to support physicians' choices in clinical fields. In particular, as it is able to extract seven different vocal anatomical structures, it can be used to study cases of *Apraxia of speech*, *Dysarthria* and to check the *velopharyngeal closure*.

According to this last problem, we want to check if the soft palate correctly touch the pharyngeal wall: a lack in this contact might cause speech's pathologies. The results are promising, since the recall is higher than the precision. In fact it is preferable to make **under-segmention** rather than over-segmention in order to treat the soft palate as separated from the pharyngeal wall even if it does not represent the ground truth. This decision is motivated by the desire to avoid false negatives and so misclassifying patients as healthy even when they actually have a pathological condition.

Bibliography

[1] *M. Ruthven, M. E. Miquel, e A. P. King*, «Deep-learning-based segmentation of the vocal tract and articulators in real-time magnetic resonance images of speech», Comput. Methods Programs Biomed., vol. 198, pag. 105814, 2021, doi: 10.1016/j.cmpb.2020.105814.

[2] *S. Firuzinia, S. M. Afzali, F. Ghasemian, e S. A. Mirroshandel*, «A robust deep learning-based multiclass segmentation method for analyzing human metaphase II oocyte images», Comput. Methods Programs Biomed., vol. 201, pag. 105946, 2021, doi: 10.1016/j.cmpb.2021.105946.

[3] *A. A. Taha e A. Hanbury*, «Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool», BMC Med. Imaging, vol. 15, n. 1, 2015, doi: 10.1186/s12880-015-0068-x.

[4] *A. R. B e Y. Wang*, «Optimizing Intersection-Over-Union in Deep», pagg. 234–244, 2016, doi: 10.1007/978-3-319-50835-1.

List of Figures

List of Tables