Academic Year 2023 – 2024
Research Unit: CREATIS
Team(s): Tomoradio
Supervisor(s): T. Baudier & D. Sarrut

I.A INSTITUT
by EPITA & ISG

CREATIS

COMPUTER SCIENCE

Automatic segmentation of salivary glands and lachrymal glands on CT scan

Benoît Catez

April 15, 2024 - June 14, 2024

Abstract

The following project was conducted as an internship project for automatic segmentation of lacrimal glands and salivary glands from CT scan using Deep Learning. Under Centre de Recherche En Aquisition et Traitement de l'Image pour la Santé (CREATIS) directions and in Centre Lèon Bérard (CLB)'s infrastructures, the project lasted two full months. It concluded with a successful model thanks to the access to Jean-zay, the supercomputer provided by the Institut du développement et des ressources en informatique scientifique (IDRIS).

You can find the github of this project here.

Contents

1 Acknowledgement

I express my gratitude to CREATIS and CLB for providing me the opportunity to work and learn in their infrastructures.

I want to offer special thanks to Thomas Baudier, my supervisor, for his support and help thorough this internship which wouldn't have concluded without his help.

I thank my team leader, David Sarrut, as well, who assisted me during my mission, tips and direction here and there.

I am also grateful to my school, IA Institut, that offers the opportunity to conduct internships.

I thank my father, who helped me for the first contact with CREATIS.

2 Introduction

2.1 CREATIS

I was an intern for CREATIS for two months, inside CLB's office. CREATIS is a multidisciplinary laboratory renowned for its expertise in medical imaging, playing a pivotal role in healthcare technologies. Specializing in magnetic resonance imaging, ultrasound, X-rays, and optics, CREATIS integrates knowledge in physics, mathematics, computer science, and instrumentation. This comprehensive expertise enables the optimization of the entire imaging process, from acquisition to image analysis and medical diagnosis assistance. Through long-standing collaborations with hospitals, CREATIS contributes significantly to the advancement of personalized and predictive medicine.

CREATIS, is a joint research unit comprising Centre National de la Recherche Scientifique (CNRS), Institut National de la Santé et de la Recherche Médicale (INSERM), Institut National des Sciences Apliquées (INSA) Lyon, Claude Bernard Lyon 1 University, and Jean-Monnet University of Saint-Etienne. With 200 members, its mission is to develop preclinical and clinical imaging methods that include:

- Expertise in multiparametric, multiscale, n-dimensional imaging, and high spatiotemporal resolution to extract anatomical, functional, and metabolic information.
- Upstream research driven by applications involving the primary research communities in acquisition (instrumentation, strategy/method), image and signal processing (simulation, modeling), and medical and health issues.
- An approach encompassing acquisition to imaging biomarker, incorporating the capability to couple various acquisition modalities.

CREATIS is structured into four research teams:

- Modeling and Analysis for Medical Imaging and Diagnosis (MYRIAD)
- NMR and Optics, From Measure to Biomarker (MAGICS)
- Ultrasound Imaging (ULTIM)
- Tomographic Imaging and Therapy with Radiation (TOMORADIO)

The laboratory boasts a wide array of specific skills in imaging, including segmentation, registration, parametric fitting and quantification, reproducibility, image reconstruction, inverse problems, distributed computing, machine learning, database structuring, radio frequency electronics, and the physics of MRI, CT scan, ultrasound, and optical modalities.

The imaging research is applied to various medical conditions such as cancer (breast, liver, prostate, colon), cardiovascular diseases, diffuse liver diseases, brain disorders (stroke, multiple sclerosis, comas, epilepsy), and acute respiratory distress syndrome. Creatis-Insa-lyon

The CLB is a hospital dedicated to cancer. As a hospital, the CLB treat patients suffering from cancer and manage them at all steps of treatment. In addition, it holds research labs to find new solutions against cancer, having partnerships with public institutions as well as private ones. Moreover, the CLB is a learning place as it delivers teaching for professionals, medical students, nurses and researchers. CLB

From the link between CREATIS and CLB, the TOMORADIO team is located within CLB's infrastructure. As part of TOMODARIO, my office was within the CLB in Lyon.

2.2 Mission

As explained on Sunaway Cancer Centre website Lu177-prostate specific membrane antigen (PSMA) therapy is a type of molecular therapy that is used to treat metastatic prostate cancer.

Lutetium-177 is a radionuclide substance that emits damaging radiation which will destroy cancer cells. It is combined with PSMA protein that seeks out PSMA receptors which are found on majority of prostate cancer cell surface. Once Lutetium-177 PSMA is infused into a patient's bloodstream, it will target and bind to PSMA receptors on prostate cancer cells. Damaging radiation from the Lutetium-177 will, over time, result in death of the prostate cancer cells.

Before commencing treatment, patients will need to undergo a PSMA PET-CT scan to confirm PSMA receptors on the cancer cells, which will be targeted by Lutetium-177 PSMA.

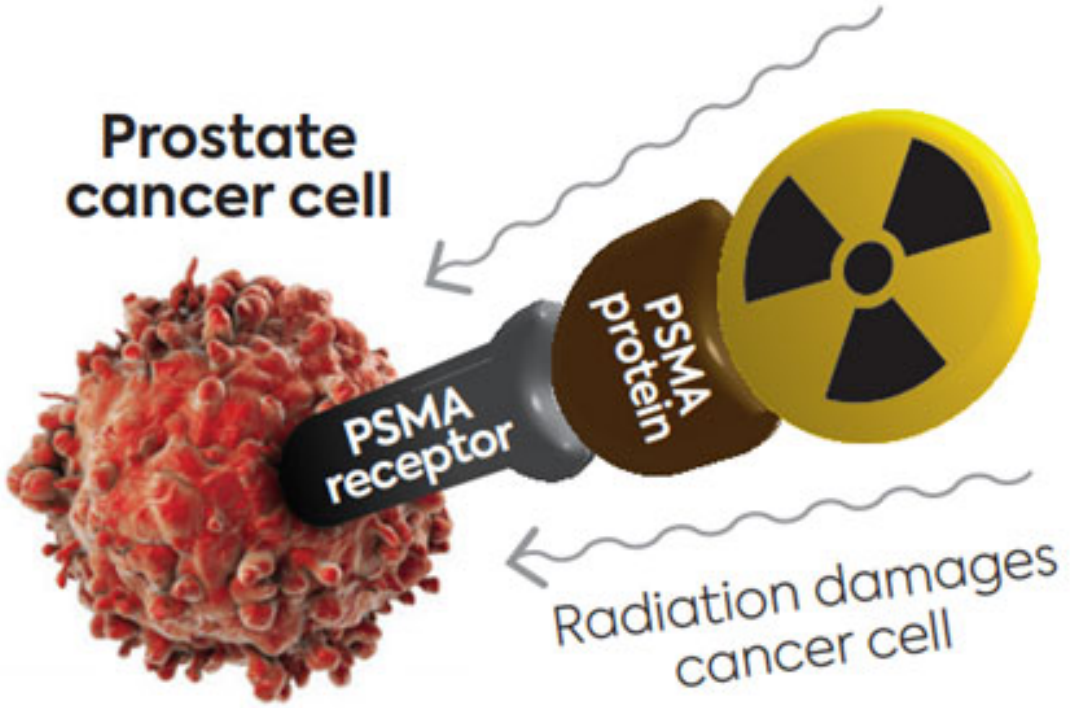

Figure 1 – Lutetium-177 PSMA Therapy, Sunaway Cancer Centre

However, prostate cancer cells are not the only ones to produce the PSMA receptors. Salivary glands and lacrimal glands produce PSMA receptors as well naturally. This is an issue regarding the dose of Lutetium-177 doctors can administrate knowing that salivary glands and lacrimal glands will be targeted as well.

As they are targeted too, some symptoms appear, such as dry eyes and mouth or lack of appetite.

To manage the administration of the dose, researchers want to use segmentation to find salivary glands and lacrimal glands within PET-CT scans. In order to achieve that, they use segmentation tools such as Total Segmentator.

Total Segmentator is a segmentation model trained on lots of body structures, such as skeleton, cardiovascular system, gastrointestinal tract, muscles and other organs (see fig-2).

Salivary glands and lacrimal glands do not appear on fig-2. Those glands are found in the head (see fig-3) but Total Segmentator does not include them.

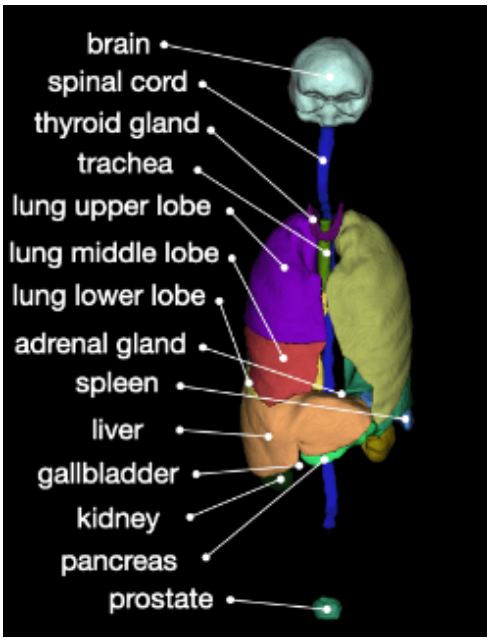

Figure 2 – Total Segmentator "other organs"

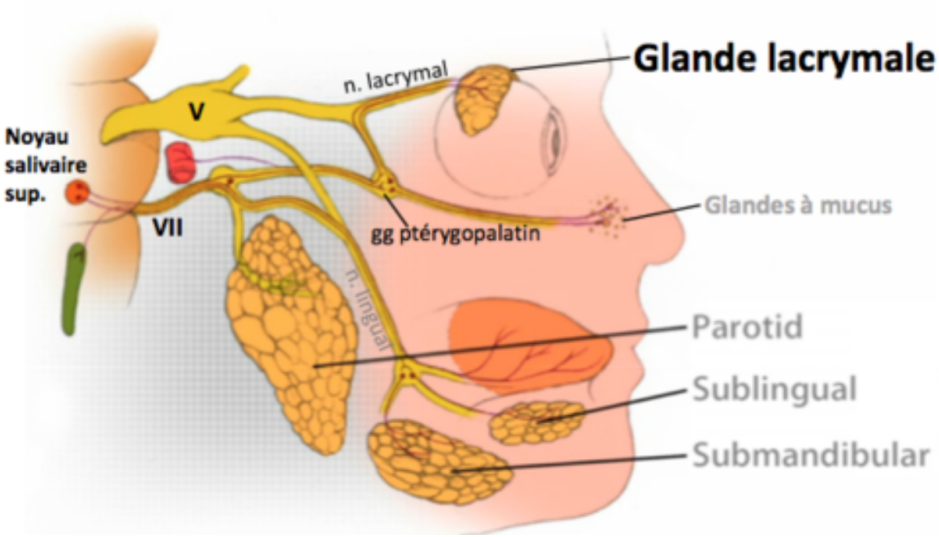

Figure 3 – Anatomical board, salivary glands and lacrimal glands

Clinical tools for segmentation exist, however, researchers cannot get easy access to those tools. They can write demand for limited access, but this is not a lasting solution.

The subject for this internship is then : **Automatic segmentation of salivary glands and lacrimal glands on CT scan**

I separated the subject in four major parts :

- Data
- Model
- Results
- Discussion

3 Project

3.1 Data

3.1.1 Original Data

This project was launched with a set of 50 patients in order to train the segmentation model with nnUNet (see 3.2.2). Each patient has a CT scan of their entire body, stored in .dcm files. Such raw data is unusable to train a segmentation model. Using nnUNet requires having data with the exact same format and size for each patient. It means ensuring that :

- size (in mm and pixel)
- spacing (in mm/pixel)
- dimension
- file format

are the same for each patient. To comprehend it, the data is visualized through an image viewer software (namely open-vv).

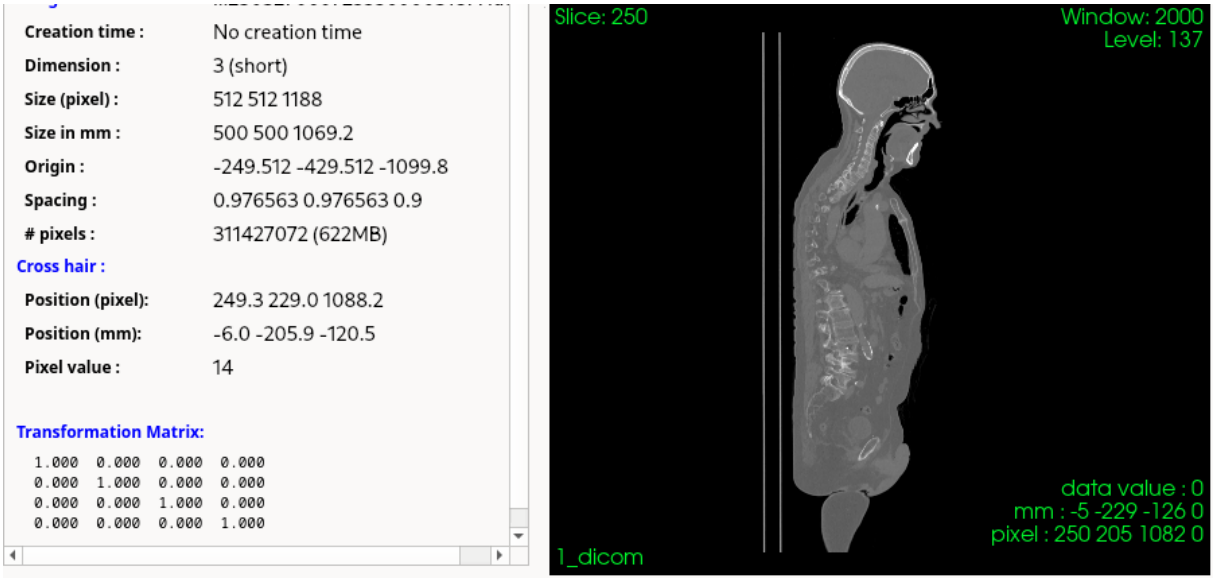

Figure 4 – Saggital view of patient 001 from the original data.

Moreover, as the training is based on supervised learning, each patient data have a label attributed to it. Each label shows segmented Region of interest (ROI) with values from 1 to 6, each value attributed to one ROI. Those are (see fig-3):

- The right lacrimal gland
- The left lacrimal gland
- The right parotid
- The left parotid
- The right sub mandibular gland
- The left sub mandibular gland

Figure 5 – Label : parotid and sub mandibular glands, coronal view, patient 001, Dataset003

Every label has been created from a clinical software that handles lacrimal glands and salivary glands.

Finally, in order to crop full bodies, skulls of patients have been extracted thanks to Total Segmentator (see fig-7).

3.1.2 Data processing

There are two transformations that must be conducted to obtain useful data :
- resampling images to be accepted by nnUNet
- cropping images to keep only ROI

Sizes are a basic need of nnUNet to perform any training with their model.

The first resampled dataset is named `"Dataset002_glands"` for convenience sake. As every patient's data have been retrieved at different times, places and with different tools and professionals. Spacings and sizes (mm and pixel) differ from a patient to another. Changes are as below :
- Spacing is set at 1:1:1 (3-dimensional as images are 3D images), so 1 pixel equal 1 mm
- size (mm or pixel) is set as the minimum of the biggest patient in each axis to get all the data. So x = 600, y = 600 and z = 2050
- Origin is fixed for every patient
- all images are saved with NIfTI file format

e.g patient 001 (see original data version on fig-4):

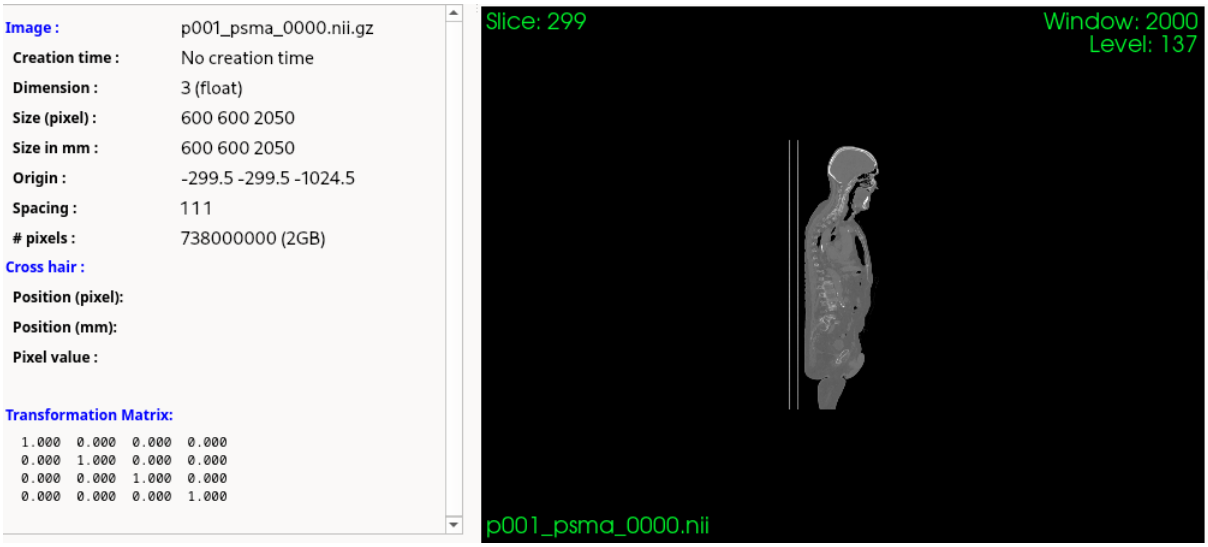

Figure 6 – Saggital view, patient 001, Dataset002

However, the data consist in full body CT scans. This is an issue for three main reasons:

- needed computing power resources
- time consumption
- amount of information the model will have analyse although most of it has nothing to do with ROI

To tackle this newfound issue, multiple methods have been discussed in order to find an easy and quick way to keep only those ROI while keeping similar images for the model to train more efficiently.

The final choice involved using Total Segmentator to use segmentations of skulls in order to locate the wanted area.

Those skulls were used as masks that perfectly correspond to their respective patient heads (see fig-7).

Figure 7 – Skull mask, saggital view, patient 001

Like CT scans, labels and skulls are resampled at the same time with the same changes so their positions continue to overlap with their CT scans.

This way, all there is to do is to find where to crop using the skulls and effectively crop the CT scans. The first step was to find the highest skull among all patients. Then define its height (plus a safety margin) as

the distance that will be cut to ensure all heads are fully taken and no data is lost.

The second step was to find the center of every patient's skull to know where the crop should take place. Then, the crop is applied on every skull, as well as their corresponding CT scan and label.

This is the creation of `"Dataset003_glands"` from Dataset002. Again, resampling makes changes :

- Spacing is set at 3:3:3 to save computer resources and processing time, so 1 pixel equal 3 mm
- size in mm changes because of the crop, so x = 600, y = 600 and z = 375
- size in pixel changes along with spacing, so as spacing is 3:3:3, size is 3 times lower than size in mm. So x = 200, y = 200, z = 125
- all images are still saved with NIfTI file format

e.g patient 001 (see dataset002 version on fig-6):

Figure 8 – Saggital view, patient 001, Dataset003

This Dataset003 was supposed to be the dataset used on the working laptop. However, it was too big for the processing power of the laptop. As the final training was planned to turn on Jean-Zay (see section-3.2.3), every training conducted on the laptop were tests to make things work. With that in mind, I created a `"Dataset004_glands"` for those tests on laptop even if it meant losing data, since this Dataset004 would not be the final one.

For this 3rd resampling from Dataset003, only CT scans and labels had a change, since the crop is already done on Dataset003, skulls are not included.

- Spacing is set at 9:9:9, so 1 pixel equal 9 mm (here some patients lose their lacrimal glands because of their small volume).
- size in mm does not change, so x = 600, y = 600 and z = 375
- size in pixel does change along with spacing, so as spacing is 9:9:9, size is about 9 times lower than size in mm. So x = 67, y = 67, z = 42
- all images are still saved with NIfTI file format

Dataset images becomes blurry so precision decreases a great amount.

e.g patient 001 (see dataset003 version on fig-8):

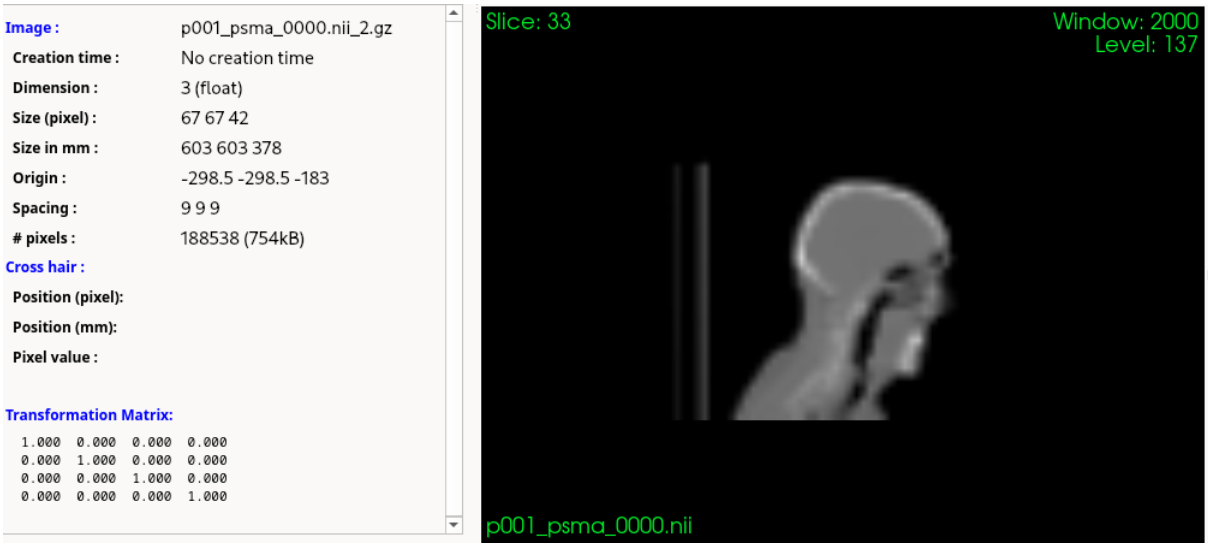

Figure 9 – Saggital view, patient 001, Dataset004

When the pipeline was ready, I created the final dataset, to conduct training bringing real results to this project.

The `"Dataset001_glands"`, made from Dataset002, used the same resampling method as Dataset003 but spacing was kept at 1:1:1 to ensure the best quality. So:

- Spacing is set at 1:1:1, so 1 pixel equal 1 mm
- size (mm and pixel) changes from the crop, so x = 600, y = 600 and z = 375
- all images are still saved with NIfTI file format

e.g patient 001 (see dataset002 version on fig-6 and dataset004 version on fig-9):

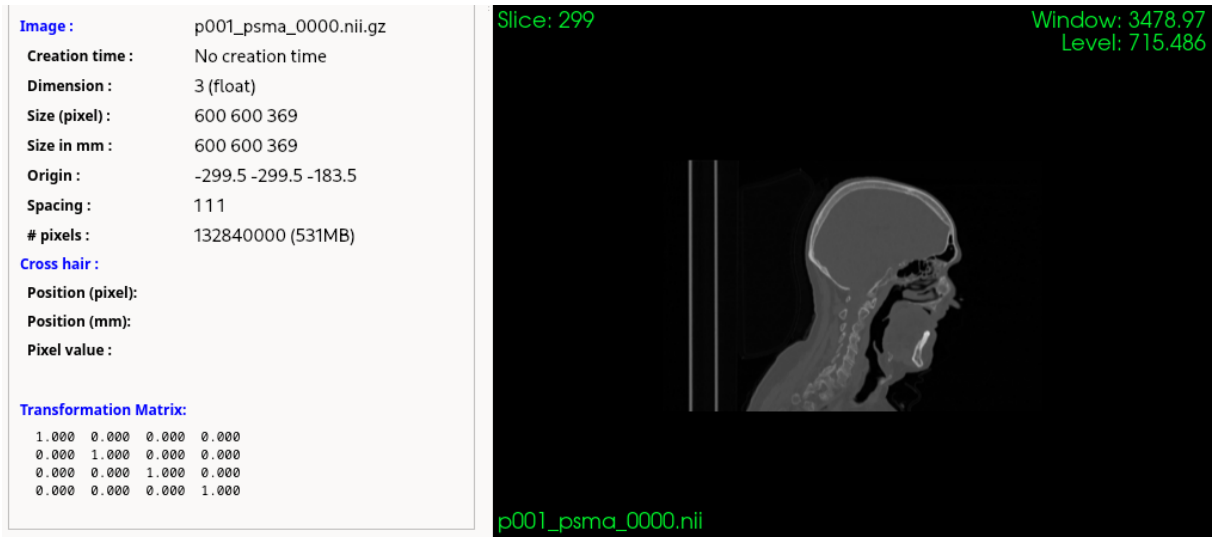

Figure 10 – Saggital view, patient 001, Dataset001

Finally, nnUNet requires a certain data architecture as well as specific file naming to automatically take data from a given environment. Data must be stored in a `nnUNet_raw` folder containing every dataset.

Every dataset must contain 2 mandatory folders and a Dataset.json file storing information about how to read the folders and labels (plus any amount of optional folders and files). Among necessary folders : "imagesTr" holds every training cases named `p[patient-id]_psma_0000.nii.gz`.

With 50 patients, 45 are stored in this folder, `10%` (so 5) is kept out to serve as validation cases. Those `10%`, randomly chosen, are stored in a "imagesTs" folder for later use.

Then a "labelsTr" folder holding every label. There, every 50 patients are present. Since every file has an ID, nnUNet will not encounter issues recognising which label corresponds to which CT scan.

The "imagesTs" folder will be ignored by nnUNet during pre-process and training.

Figure 11 – `nnUNet_raw` architecture

3.1.3 Data analysis

One of the most important part of this type of project is to know your data. Hence, the first step is to find anomaly among patients. There exist one anomaly in patient 009 : the top of his head is missing. It does not make any ROI disappear but it is important to take it in consideration.

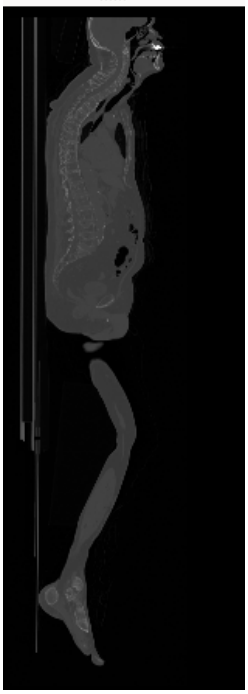

Figure 12 – Saggital view, patient 009, Original data

Then, it is vital to see some characteristics of the data.

Note: The green shade on graphs represents the 5 cases from "imagesTs", the intent here is to see if those 10% are representative of the entire dataset.

The first visualisations are based on volume of ROI. Figures 22, 23 and 24 in Annex(5) represent distribution of volume (in mm^3) for each ROI, left and right. Left and right for each glands have similar volume distribution and validation images follow approximately the whole shape.

The following visualisations are based on HU. You can use this table for reference.

As seen in figure 25, most of pixels are either air or background (background is set as minimum value by default). This is unusable, so for figure 26, which is sensibly the same graph, useless values were cut. Anything below fat density and anything over bone density is removed. As expected, most of the head is blood, muscle and gray matter.

The next visualisations are a close up on patient001. On figures : 27, 28, 29, 30, 31 and 32, distribution between left and right is extremely similar. However, density is lighter in parotids than in lacrimal glands and sub mandibular glands that are looking alike.

The last visualisations are distributions of mean HU for each patient of each ROI. From figures 33, 34, 35, 36, 37 and 38, we can see that validation cases follow the trend while not matching it exactly.

3.2 Model

3.2.1 U-Net

This project uses U-Net to bring results. A U-net is a type of CNN primarily designed for image segmentation tasks, such as biomedical image segmentation. The architecture was introduced by Olaf and Thomas in their paper "U-Net: Convolutional Networks for Biomedical Image Segmentation.".

The U-net architecture is characterized by a U-shaped structure, which consists of two main parts: the contracting path (encoder) and the expansive path (decoder).

- Contracting Path (Encoder):
  - Convolutions: The contracting path follows the typical architecture of a convolutional network. It consists of a series of convolutional layers, each followed by a ReLU activation function and a pooling layer. This path captures the context of the image
  - Max Pooling: After each convolutional block, max pooling layers reduce the spatial dimensions of the feature maps, effectively performing down-sampling. This process captures the hierarchical features of the input image.
- Bottleneck : At the bottom of the U, the network contains a bottleneck layer that represents the most compressed representation of the input. This layer typically involves more convolutions and captures the deepest features before the network starts to expand again.
- Expansive Path (Decoder)
  - Up-Convolutions (Up-sampling): The expansive path consists of up-convolution (or transposed convolution) layers that increase the spatial dimensions of the feature maps. This path reconstructs the image from the encoded features.
  - Skip Connections: A distinctive feature of the U-net is the presence of skip connections. These connections link the corresponding layers in the contracting path to the expansive path, allowing the network to use high-resolution features from the encoder to improve the up-sampling process in the decoder. This helps in retaining fine details and improves the segmentation quality.

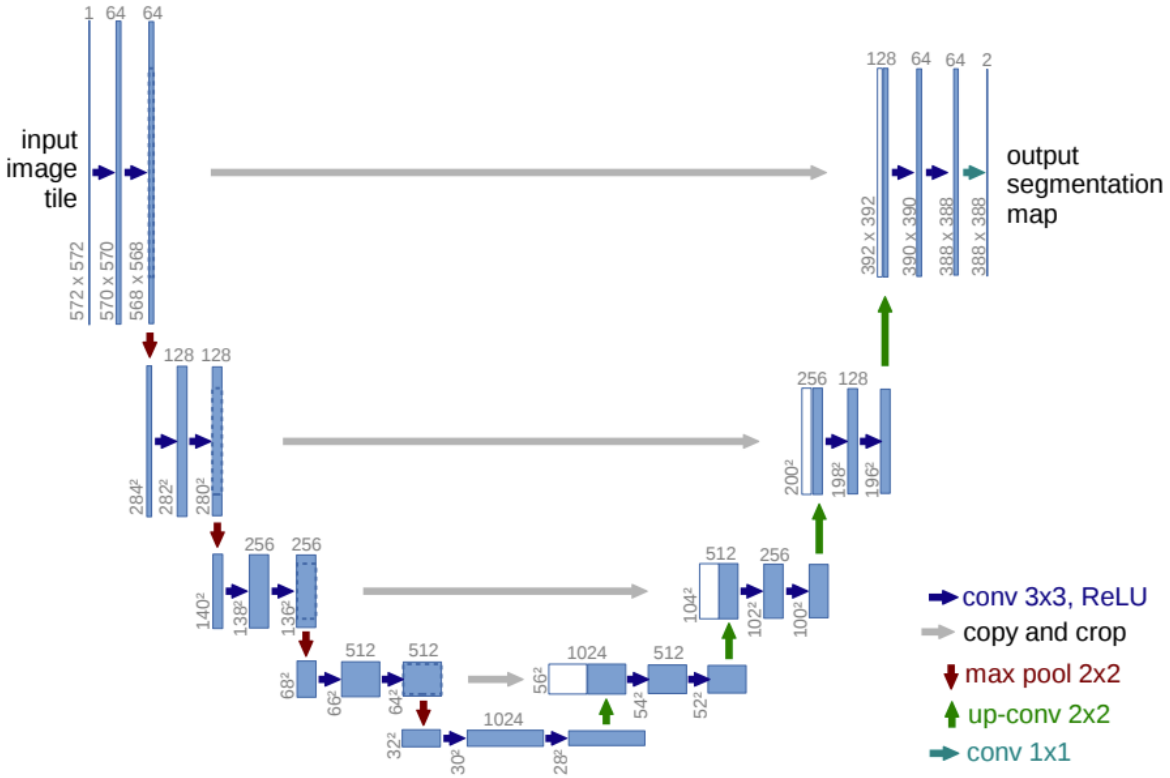

Figure 13 – U-Net CNN

Detailed Layer Configuration :

- Input Layer: Takes an image of size (H, W, C), where H is the height, W is the width, and C is the number of channels (only one in our case)
- Convolutional Blocks: Each block consists of two 3x3 convolutions followed by a ReLU activation and a 2x2 max pooling.
- Down-sampling Path: Series of convolutional blocks with pooling, typically doubling the number of feature channels after each block.
- Bottleneck: The final convolutional block before up-sampling.
- Up-sampling Path: Each step consists of an up-convolution followed by a concatenation with the corresponding feature map from the down-sampling path via skip connections, followed by two 3x3 convolutions and ReLU activations.
- Output Layer: A final 1x1 convolution reduces the number of channels to the desired number of output classes for segmentation.

Lecun et al.'s seminal work on convolutional networks laid the groundwork for the development of the U-net architecture. In his 1998 paper "Gradient-Based Learning Applied to Document Recognition," LeCun and his colleagues introduced the concept of convolutional layers, pooling layers, and backpropagation in CNNs.

LeCun demonstrated how convolutional layers can automatically learn spatial hierarchies of features from images. This principle is extensively used in the U-net architecture for both the contracting and expansive paths.

His work showed the effectiveness of pooling layers in reducing the spatial dimensions while retaining essential features, a technique crucial for the down-sampling path in U-net.

The development of efficient backpropagation methods for training CNNs is fundamental to training U-nets, which also rely on gradient-based optimization techniques.

The hierarchical feature extraction mechanism proposed by LeCun is a core concept used in U-nets to capture both local and global features at multiple scales.

In summary, U-net builds upon the foundational principles of Convolutional Neural Network established by Lecun et al., extending them with a unique architecture designed for precise image segmentation tasks. The combination of down-sampling and up-sampling paths, along with skip connections, enables U-net to produce high-resolution segmentation maps essential for applications such as medical imaging.

3.2.2 nnUNet

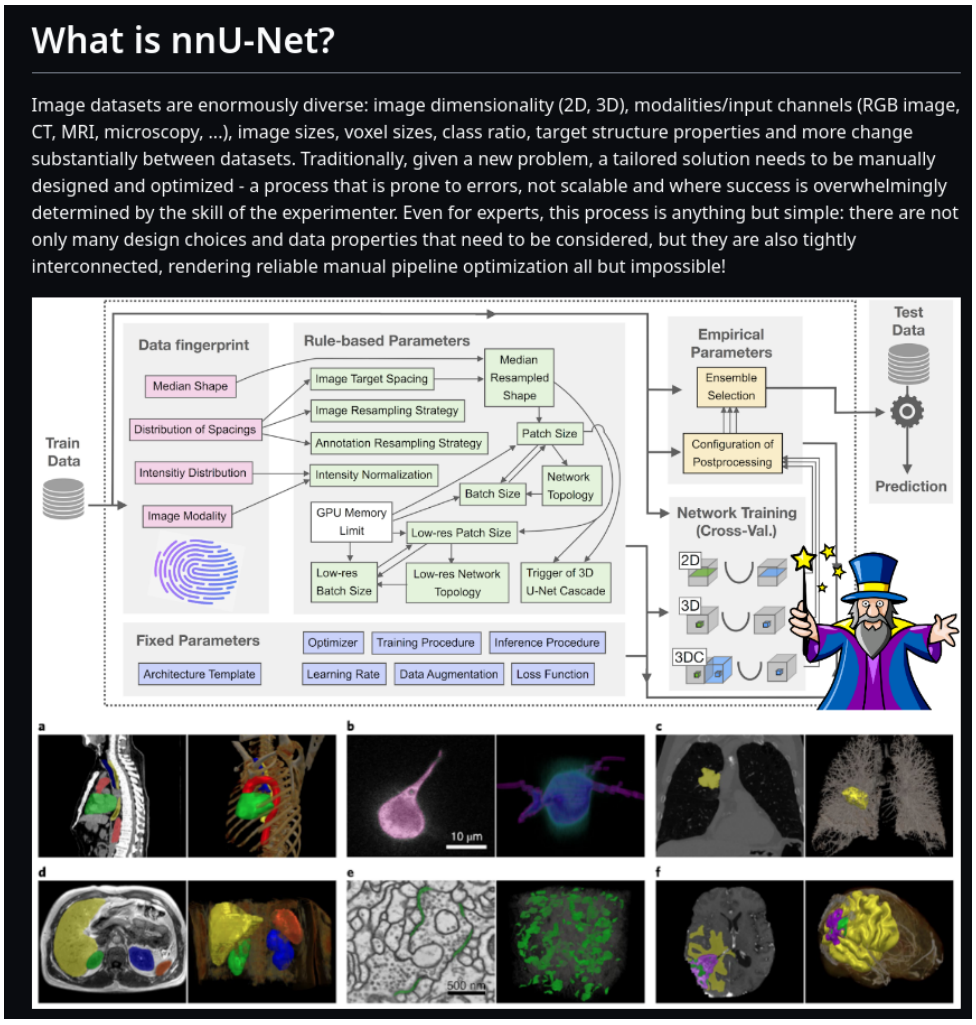

Figure 14 – nnUNet github description

The nnUNet is a segmentation method that automates the configuration of a U-Net-based segmentation pipeline tailored to a given dataset, making it user-friendly and eliminating the need for expertise. Indeed, my implication in the model itself was minimal. Evaluated on 23 biomedical datasets, nnUNet has demonstrated outstanding performance, winning multiple competitions and being widely used as a baseline in segmentation challenges.

nnUNet is designed for domain scientists and AI researchers. For domain scientists, it offers an out-of-the-box solution for image analysis, ensuring excellent results with minimal effort. For AI researchers, it provides a competitive baseline algorithm and a method development framework, facilitating the testing and improvement of segmentation methods.

The capabilities of nnUNet are extensive. It can handle both 2D and 3D images with varying modalities

and channels, and it is robust against voxel spacings, anisotropies, and class imbalances. It requires training cases for supervised learning, processing images ranging from 40x40x40 up to 1500x1500x1500 pixels in 3D and 40x40 up to 30000x30000 pixels in 2D.

The configuration process of nnUNet involves creating a 'dataset fingerprint' from the training cases. Based on this analysis, it generates several U-Net configurations: a 2D U-Net for both 2D and 3D datasets, a 3D full-resolution U-Net for 3D datasets, and a 3D low-resolution followed by a 3D cascade full-resolution U-Net for large 3D datasets, though the latter is omitted for small image sizes. My actual trainings were 2D and 3D full resolution, 2D being tests only while 3D full resolution was the final training as well.

nnUNet employs three types of parameters: fixed, rule-based, and empirical. Fixed parameters use a robust default configuration, including the loss function (did not have access to it, did not find what type of loss function it was), data augmentation, and learning rate. Rule-based parameters adapt to the dataset fingerprint using heuristic rules, optimizing network topology, patch size, and batch size. Empirical parameters are determined through trial-and-error, identifying the best U-Net configuration and post processing strategy.

nnUNet excels in scenarios requiring training from scratch, such as non-standard image modalities, biomedical challenge datasets, and 3D segmentation problems. However, it is not ideal for standard 2D RGB image segmentation problems like cityscapes, where pre-trained foundation models, which offer better initialization from large datasets, perform better. Additionally, nnUNet does not support foundation models due to their limited utility in non-standard settings, 2D architecture constraints, and incompatibility with nnUNet's design principle of adapting network topology to each dataset.

As issues arose, i happened to wander in nnUNet's code to resolve them. While none of the issues were solved thanks to this exploration, my understanding in the system widened.

In the end, while nnUNet did not allow me to meddle with the development of the model, it made possible completing the entire mission within the given 2 months.

3.2.3 Jean-Zay

Jean Zay is the new converged supercomputer acquired by the Ministry of Higher Education, Research, and Innovation through the company GENCI (Grand Équipement National de Calcul Intensif). The acquisition contract, awarded to Hewlett-Packard Enterprise (HPE), was signed on January 8, 2019. It was installed at the IDRIS, national calculation center on the CNRS.

It holds more than 3000 GPUs and is available for free, as long as you come with a real project and that your identity is verified.

I was able to connect to Jean-Zay's server through my project and was allocated with 7000 hours of runtime (I used less than 200 hours for the record) that i could use by starting Jobs. To launch a Job, i needed to put my commands in a special file to set up running parameters such as running environment, resource allocation, time limit for the Job and an output file to have feedback on what happened during the Job.

While having a familiar interface, a command terminal, it was also tremendously faster than my laptop. To illustrate, I compared the training time for a 5 epochs run on dataset004 (see fig-9) on my laptop and on Jean-Zay. With one GPU, my laptop took 1 hour and 15 minutes while Jean-Zay finished it in 1 minute (all configurations where the same).

Figure 15 – Jean-Zay supercomputer

3.2.4 Metric - Dice

To evaluate my model, I used Dice as a metric. The principle behind this measure is the similarity between two elements.

In my case, i measure the degree of overlapping of ROI's volume, from the formula :

$$Dice = \frac{2|V_1 \cap V_2|}{|V_1| + |V_2|}$$

So Dice value is between 0 (no overlapping at all) and 1 (two volumes are the same).

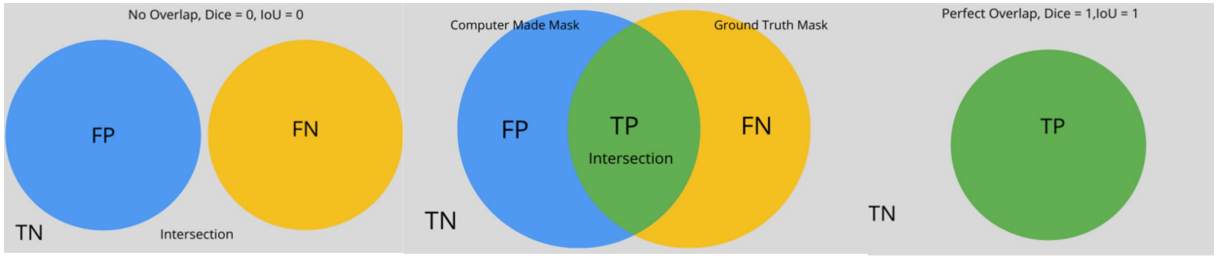

Figure 16 – What is dice, 3 states

(on fig-16
- FP = False Positive
- FN = False Negative
- TP = True Positive
- TN = True Negative)

3.3 Results

3.3.1 Training

Thanks to nnUNet (see section 3.2.2) training a model is extremely simple. When all training cases and labels are stored in the right way (see fig-11), you can start the "plan and preprocess". At this point nnUNet will automatically check the integrity of the given dataset, then prepare all the parameters and hyper-parameters for all training configurations (only specified ones if needed).

This step asks for a lot of computer memory. So much that my laptop was only able to plan and preprocess dataset004 and dataset003. I needed Jean-Zay to go through dataset001 (in multiple step, because the plan and preprocess command is a fusion of 3 smaller commands, and specifying one configuration, to avoid taking unnecessary memory) while, even in multiple steps, dataset002 never got through this part even using Jean-Zay.

```
├── nnUNet_preprocessed
│   ├── Dataset004_glands
│   │   ├── dataset.json
│   │   ├── dataset_fingerprint.json
│   │   ├── nnUNetPlans.json
│   │   ├── splits_final.json
│   │   ├── gt_segmentations
│   │   │   ├── p001_psma.nii.gz
│   │   │   ├── p002_psma.nii.gz
│   │   │   └── ..
│   │   ├── nnUNetPlans_2d
│   │   │   ├── p001_psma.npy
│   │   │   ├── p001_psma.npz
│   │   │   ├── p001_psma.pkl
│   │   │   ├── p001_psma_seg.npy
│   │   │   └── ..
│   │   └── nnUNetPlans_3d_fullres
│   │       ├── p001_psma.npz
│   │       ├── p001_psma.pkl
│   │       └── ..
```

Figure 17 – `nnUNet_preprocessed` architecture

Even if nnUNet is automatically defining most of parameters and hyper-parameters, as a user, you can specify some of them. For example, nnUNet will make, by default, a 5 fold cross validation. You can change, to a certain extent, specifics about it (for the record, i used default for cross validation).

Then you can start the training right away. You need to specify :

- used dataset - required
- configuration (e.g 3d full resolution) - required
- epoch number (default is 1000 epochs) - optional
- specific trainer (e.g run a benchmark) - optional

When the training starts, nnUNet will save a checkpoint at every epoch (replaced each time) so you can stop the training and resume it at a later time if needed.

I mostly used dataset004 to test configurations and trainers. I did several benchmarks, used 2d and 3d full resolution and specified different epoch amounts.

Then, for the final training, i used dataset001 (see fig-10). So minimum spacing while keeping only the head for each patient.

I chose to go for 250 epochs with the 3d full resolution. For about 1 minute and 30 seconds per epoch, the training ran for about 6 hours and 15 minutes in total.

There appeared a major issue that did not happen with dataset004 during tests. The predictions, while looking very good at first glance, were all messed up. nnUNet seemed to have issues differentiating right and left glands, and even, for some patients, up and down (parotid instead of sub mandibular glands (see fig-3)).

I figured the data augmentation should be the culprit. Indeed, among the data augmentation methods, one called "Mirroring" caught my attention.

As different organs, each glands need to be segmented separately. Like lungs, both are lungs but one is smaller than the other one. So you cannot consider the left one the same as the right one. They need to be considered two different organs instead of a pair of one organ.

It is the same for salivary glands and lacrimal glands. I cannot accept a data augmentation method that will make new images with symmetrical modifications.

Data augmentation is not something that can be meddle with when using nnUNet. If you need to for some reason, then you need to change the inner program of nnUNet and make your own version of nnUNet. Not something that i could do within 2 months. Luckily, maybe, as nnUNet is focused on medical segmentation tasks, the case when organs should not be augmented with symmetrical changes often happens.

I found an option that turns off this "Mirroring" data augmentation method. So i trained my model again and had positive results.

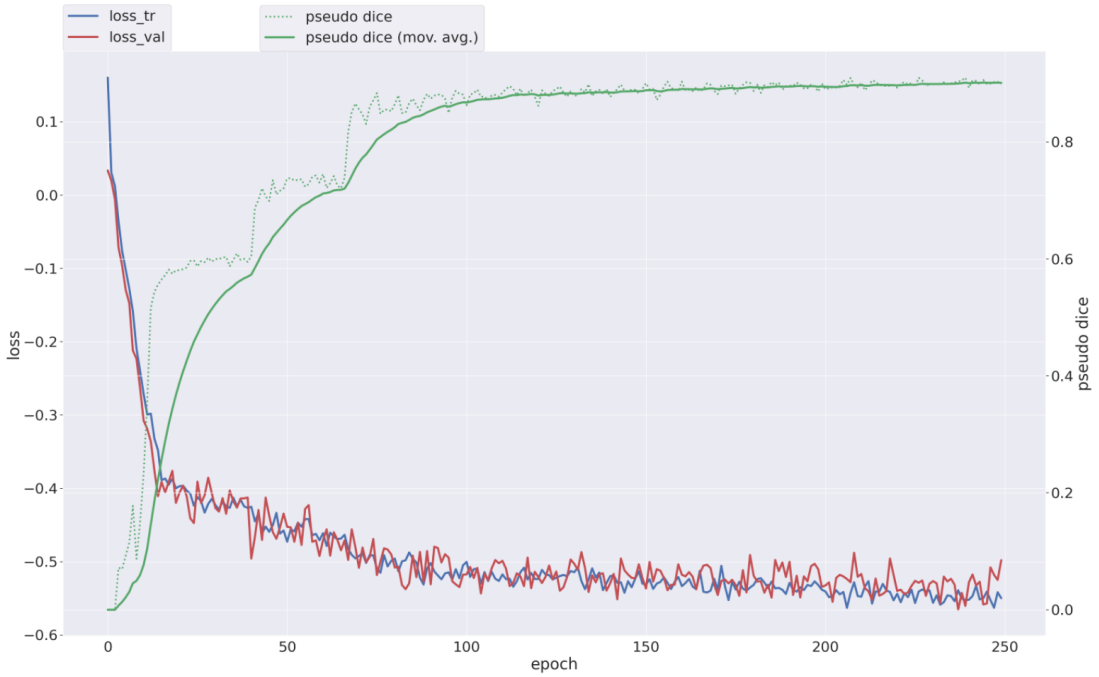

Figure 18 – Training results : dataset001, 250epochs, 3d full resolution

As my loss converged, i can say that going for more epochs would have been a wasted of computer resources. And the average dice after the training is 0.9.

Those results are stored in a `nnUNet_results` folder with a similar architecture as "raw" and "preprocessed".

```
├─nnUNet_results
|  ├─ Dataset004_glands
|  |  ├─ nnUNetTrainer_nnUNetPlans_3d_fullres
|  |  |  ├─ dataset.json
|  |  |  ├─ dataset_fingerprint.json
|  |  |  ├─ plans.json
|  |  |  ├─ fold_0
|  |  |  |  ├─ checkpoint_best.pth
|  |  |  |  ├─ debug.json
|  |  |  |  ├─ progress.png
|  |  |  |  ├─ training_log_[date+hour+sec].txt
|  |  |  |  └─ ..
|  |  |  └─ ..
|  |  └─ ..
|  └─ ..
```

Figure 19 – `nnUNet_results` architecture

3.3.2 Inference

Inference consist in passing data that the model never saw for the training through the model. It is used to evaluate performance.

In order to conduct it, i randomly chose 5 patients (10% of the dataset) that are stored in the "imagesTs" folder that was ignored by nnUNet during the training. The list of patients contained patients 12, 16, 21, 22 and 47.

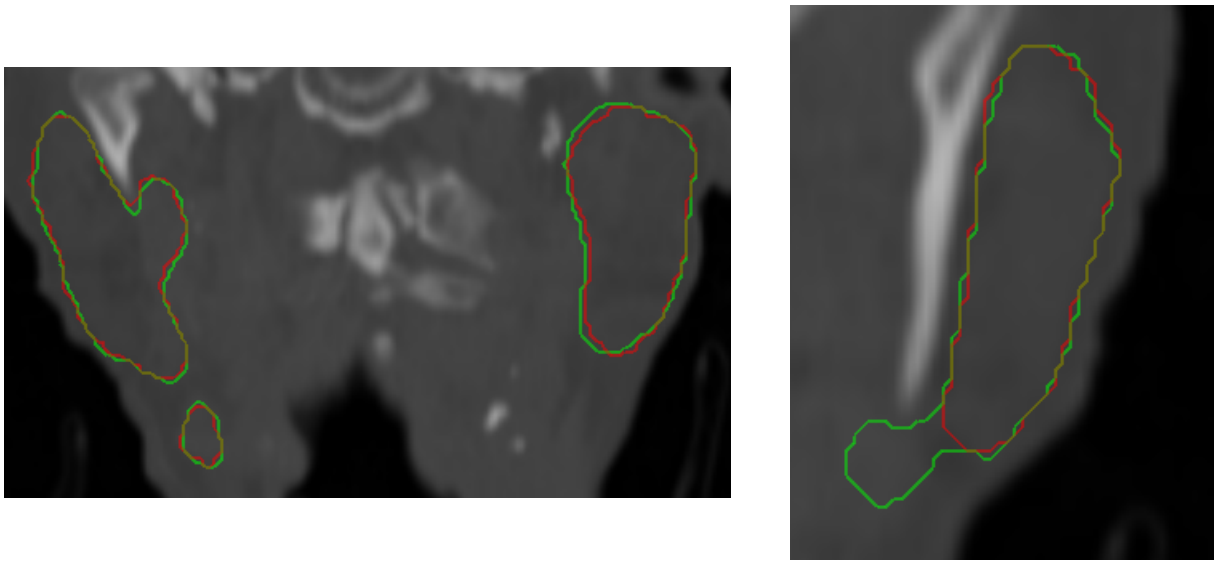

Figure 20 – Patient012, segmentation output. Inference = green, label = red

The results are quite good. On the left side of figure 20, the predicted segmentation is following the label with good precision. However, the left side shows that the model is making some mistakes even with the average dice of 0.899 for the inference.

I also used mean HU of every ROI to compare inference segmentation predictions to their labels. From figures 41, 42 and 43, we can see that parotid and the left sub mandibular gland are more accurately predicted than on figures 39, 40 and 44 that are more approximate. As lacrimal glands are smaller than salivary glands, a similar gap in precision will have a greater impact on the overall accuracy of lacrimal glands' segmentation.

3.4 Discussion

In the end, this project brings positive results overall. A dice measuring about 90% accuracy with a converging training.

However, this project is lacking in several aspects. Indeed, for starters, dataset size is insufficient. 50 patients (minus 5 for inference purposes) is clearly not enough not create a robust and accurate model.

For the same reason, it is difficult to have a significantly representing validation part for inference with this few data.

Then, the overall time frame to create such project in real depth is underwhelming. Even if the objective has been reached, the project will stay as a quality draft in order to be used and further developed with more time and in depth thinking.

Additionally, the model only uses dice (see section 3.2.4) as a metric to evaluate its accuracy. More metrics should reinforce the model. For example, the correction time should be meaningful.

4 Conclusion

To conclude, I started by understanding my tools for this mission.

Then I created datasets in order to adapt my data to nnUNet requirements and limits of my computer. I had to rethink the process when it occurred that using full body CT scans would be too heavy and inefficient. I needed to crop images to attain a satisfying starting point for preprocessing and training.

Next came plans and preprocess, automatically done by nnUNet, yet a arduous task to understand what it did and what could be modified. Nonetheless a very useful tool.

Jean-Zay was a wonderful device to utilize to train models. Having strict rules to follow, the tiniest clog in the system could block an entire program. But it was easy to manage overall thanks to its similarity with basic command terminals.

Finally, the model brought good results. Even if its accuracy is somewhat lacking sometimes and that those results are evaluated only by dice measure.

Additionally, this internship made me realise the extent a relatively big project can have. As well as the fact that things never fully follow plans, so how arduous of a task keeping a project within its deadline can be.

Also, applying the ReCoCo (Report, Consult, Communicate) method, turned out to make decision making when issue arise rather smooth.

This internship has been an extremely valuable opportunity for my growth.

5 Annex

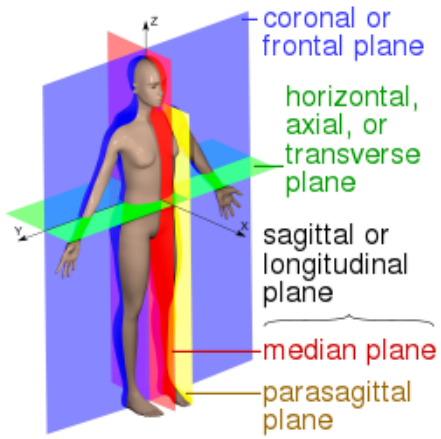

Figure 21 – Wikipedia contributors [2024b]

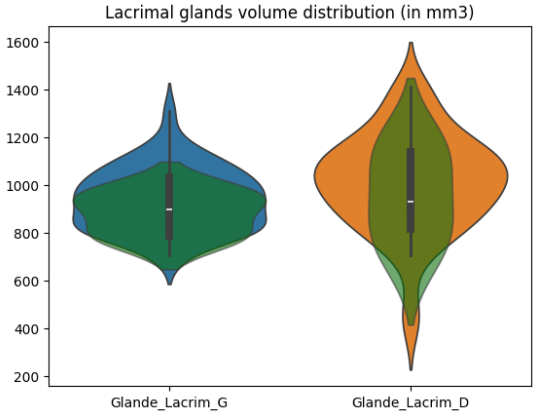

Figure 22 – Volume distribution of lacrimal glands

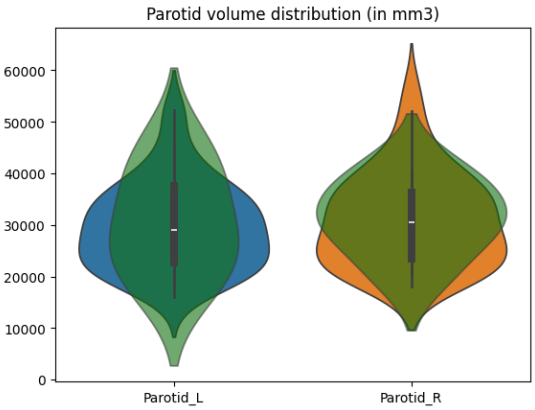

Figure 23 – Volume distribution: parotids

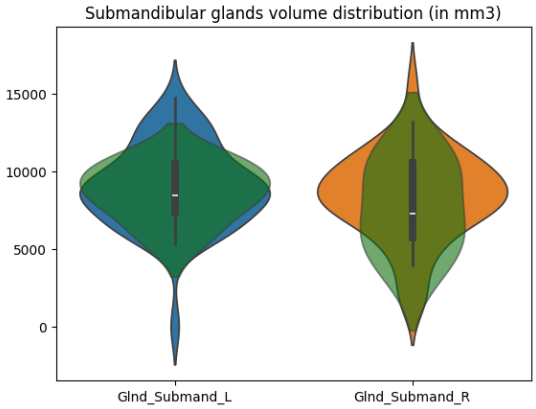

Figure 24 – Volume distribution : sub mandibular glands

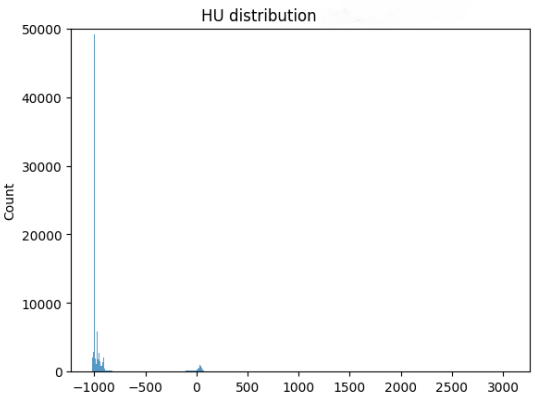

Figure 25 – HU distribution

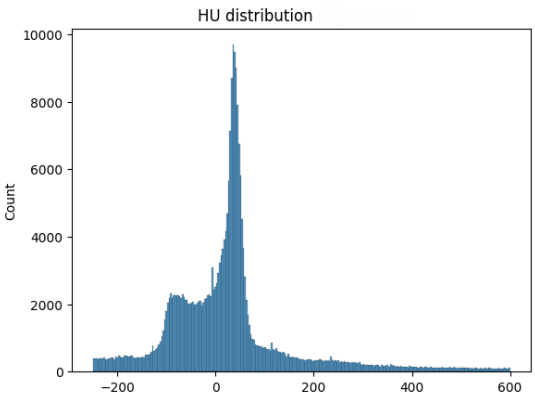

Figure 26 – HU distribution (Focused HU)

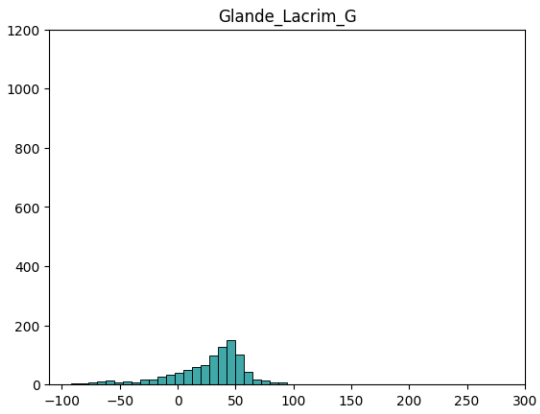

Figure 27 – Left lacrimal gland, HU distribution patient001 (x axis is HU)

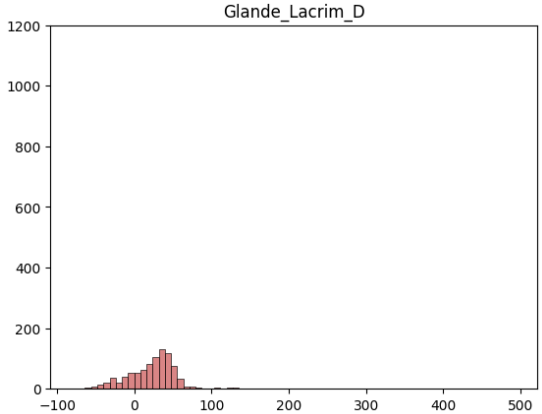

Figure 28 – Right lacrimal gland, HU distribution patient001 (x axis is HU)

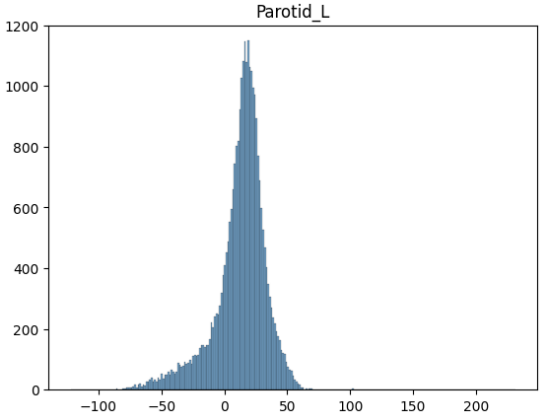

Figure 29 – Left parotid, HU distribution patient001 (x axis is HU)

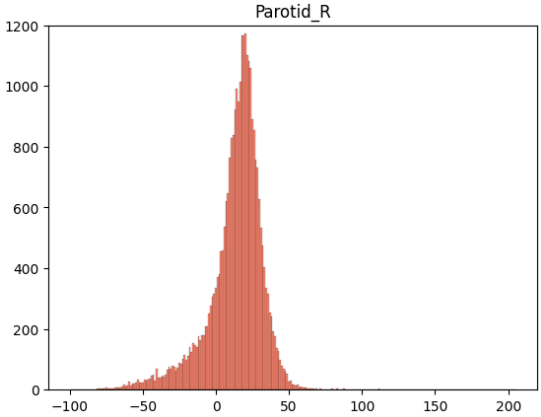

Figure 30 – Right parotid, HU distribution patient001 (x axis is HU)

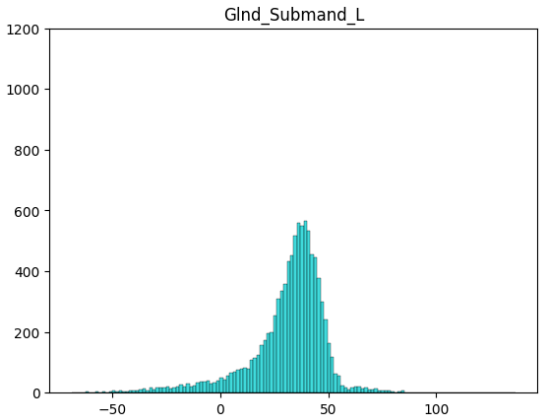

Figure 31 – Left sub mandibular gland, HU distribution patient001 (x axis is HU)

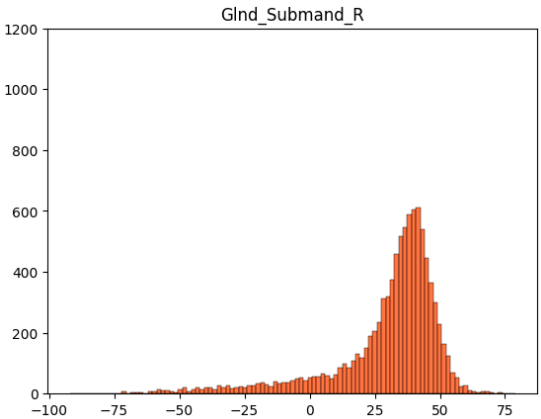

Figure 32 – Right sub mandibular gland, HU distribution patient001 (x axis is HU)

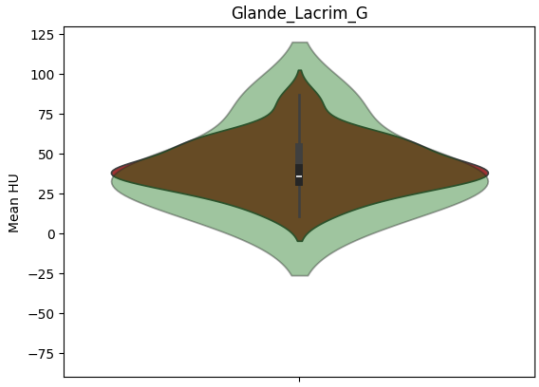

Figure 33 – Left lacrimal gland,mean HU distribution

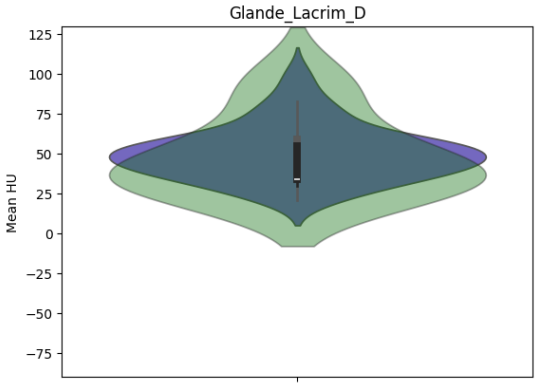

Figure 34 – Right lacrimal gland,mean HU distribution

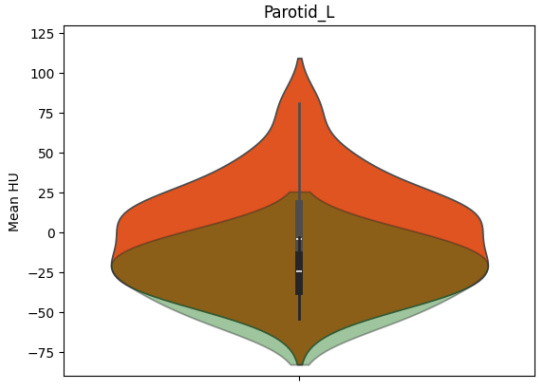

Figure 35 – Left parotid, mean HU distribution

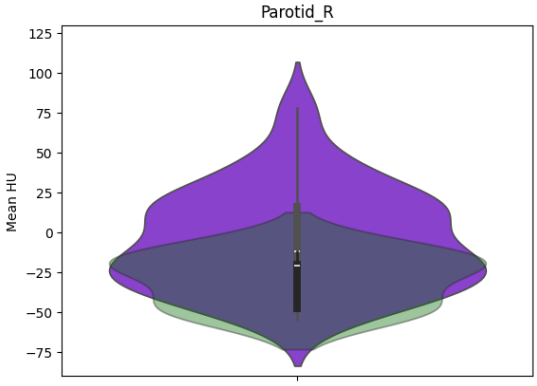

Figure 36 – Right parotid, mean HU distribution

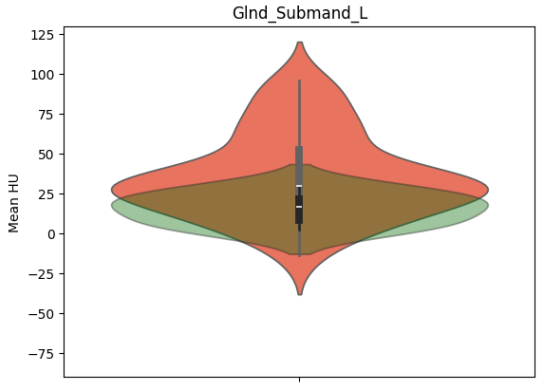

Figure 37 – Left sub mandibular gland, mean HU distribution

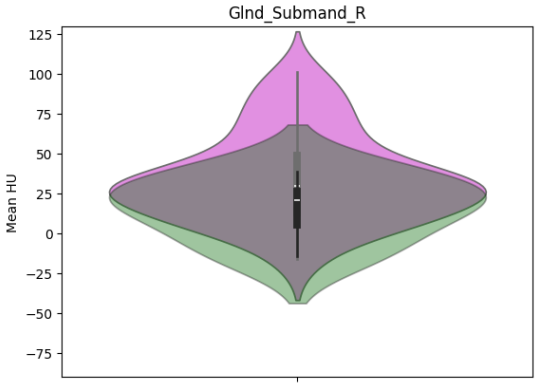

Figure 38 – Right sub mandibular gland, mean HU distribution

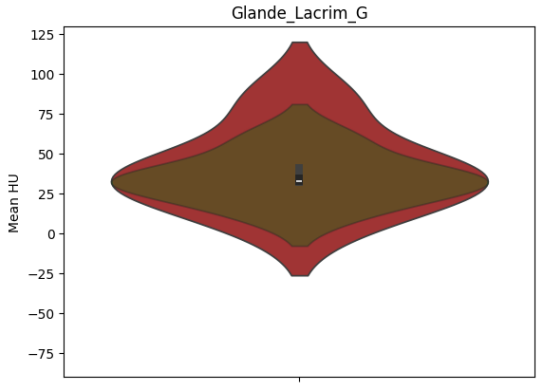

Figure 39 – Left lacrimal gland,mean HU distribution label / inference comparison

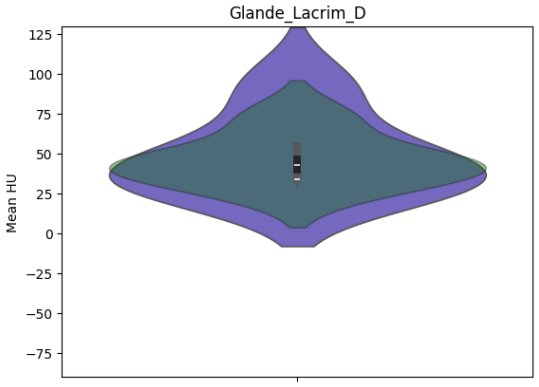

Figure 40 – Right lacrimal gland,mean HU distribution label / inference comparison

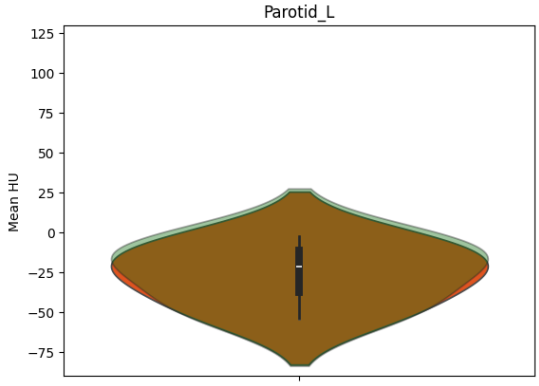

Figure 41 – Left parotid, mean HU distribution label / inference comparison

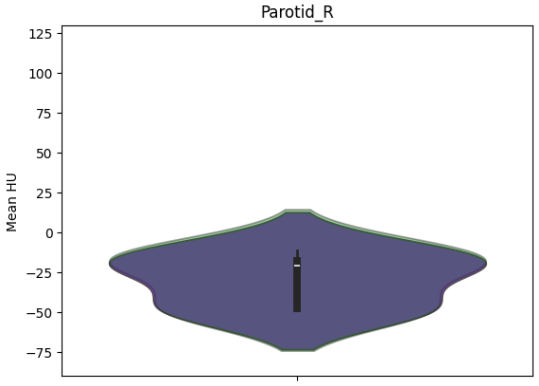

Figure 42 – Right parotid, mean HU distribution label / inference comparison

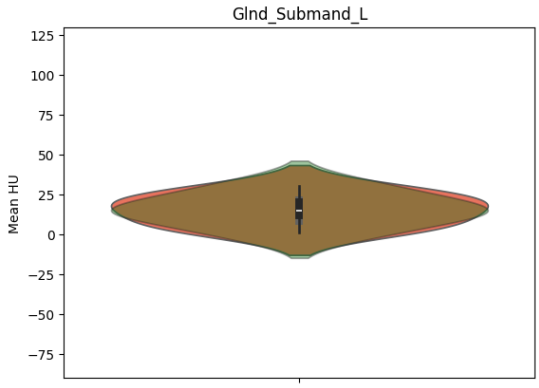

Figure 43 – Left sub mandibular gland, mean HU distribution label / inference comparison

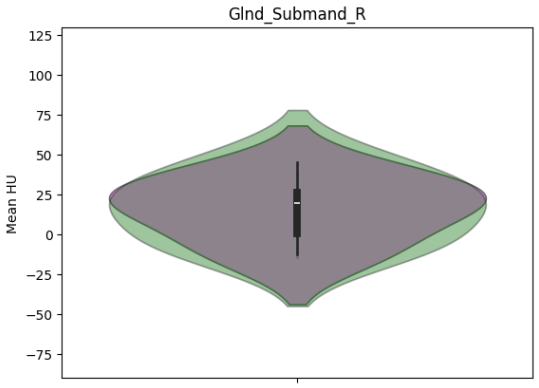

Figure 44 – Right sub mandibular gland, mean HU distribution label / inference comparison

Glossary

.dcm Digital Imaging and Communications in Medicine (DICOM) is a technical standard for the digital storage and transmission of medical images and related information.It includes a file format definition, which specifies the structure of a DICOM file (.dcm), as well as a network communication protocol that uses TCP/IP to communicate between systems. The primary purpose of the standard is to facilitate communication between the software and hardware entities involved in medical imaging, especially those that are created by different manufacturers.The DICOM standard has been widely adopted by hospitals and the medical software industry, and is sometimes used in smaller-scale applications, such as dentists' and doctors' offices Wikipedia contributors [2024d]. 7

convolution Convolution is a mathematical operation that allows the merging of two sets of information. In the case of CNN, convolution is applied to the input data to filter the information and produce a feature map. This filter is also called a kernel, or feature detector. To perform convolution, the kernel goes over the input image, doing matrix multiplication element after element. The result for each receptive field (the area where convolution takes place) is written down in the feature map. We continue sliding the filter until the feature map is complete. Gavrilova [2021]. 13, 14

coronal Plane in space corresponding to the front of a body (see fig-21). 8

correction time Time needed for a professional to correct model mistakes and/or lack of accuracy. Can be used as a metric to evaluate if a model is worth using more than another or if professionals would take less time to segment themselves.. 20

cross validation One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce variability, in most methods multiple rounds of cross-validation are performed using different partitions, and the validation results are combined (e.g. averaged) over the rounds to give an estimate of the model's predictive performance.Wikipedia contributors [2024c]. 18

CT scan A procedure that uses a computer linked to an x-ray machine to make a series of detailed pictures of areas inside the body. The pictures are taken from different angles and are used to create 3-dimensional (3-D) views of tissues and organs. A dye may be injected into a vein or swallowed to help the tissues and organs show up more clearly. A computed tomography scan may be used to help diagnose disease, plan treatment, or find out how well treatment is working Definition CT scan. 1, 4–7, 9, 10, 12, 21, 26

data augmentation Method used to artificially increase dataset sizes. In the case of images, it can be changing colors slightly, small rotations, modifying contrasts, using crops and so on, to create other images to feed the model.. 16, 18, 19

Deep Learning Deep learning is a subset of machine learning that uses multi-layered neural networks, called deep neural networks, to simulate the complex decision-making power of the human brain Definition Deep Learning. 1

epoch In machine learning, an epoch refers to one entire passing of training data through the algorithm. When at least every data has gone through the algorithm, the epoch changes and starts again until the last one is complete. 16

HU The Hounsfield scale, is a quantitative scale for describing radiodensity. It is frequently used in CT scans, where its value is also termed CT number. 0 is water and air corresponds to -1000. Wikipedia contributors [2024a]. 13, 20

Job A job is a set of command hosted on a server you can connect your computer to. That enables programs to run in parallel of your computer without pumping resources from it.. 16

label In machine learning, data labeling is the process of identifying raw data (images, text files, videos, etc.) and adding one or more meaningful and informative labels to provide context so that a machine learning model can learn from it. For example, labels might indicate whether a photo contains a bird or car, which words were uttered in an audio recording, or if an x-ray contains a tumor. Data labeling is required for a variety of use cases including computer vision, natural language processing, and speech recognition. What is data labeling. 7–10, 12, 18, 20, 25

lacrimal gland A gland that secretes tears. The lacrimal glands are found in the upper, outer part of each eye socket Definition lacrimal glands. 1, 5–8, 10, 13, 19, 20, 22–25

mask A segmentation mask is a specific portion of an image that is isolated from the rest of an image. You can use the output of a segmentation mask to copy exact areas of an image that have been assigned a label in a computer vision model. Jay Lowe, How to Create Segmentation Masks with Roboflow. 9

NIfTI The Neuroimaging Informatics Technology Initiative (NIfTI) is an open file format commonly used to store brain imaging data obtained using Magnetic Resonance Imaging (MRI) methods. Wikipedia contributors [2024g]. 8, 10, 11

resampling Changing the pixel dimensions of an image is called resampling. Resampling can degrade image quality. Downsampling decreases the number of pixels in the image, while upsampling increases the number. Resample. 8, 10, 11

saggital Plane in space corresponding to the side of a body (see fig-21). 7, 9–12

salivary gland A gland in the mouth that produces saliva Definition salivary glands. 1, 5, 6, 8, 19, 20

segmentation In digital image processing and computer vision, image segmentation is the process of partitioning a digital image into multiple image segments, also known as image regions or image objects (sets of pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics Wikipedia contributors [2024e]. 4–7, 9, 13–16, 19, 20

spacing Distance between the center of pixels in images (2D or 3D), each axis may have different spacing. 7, 8, 10, 11, 16, 18

supervised learning Supervised learning is a paradigm in machine learning where input objects (for example, a vector of predictor variables) and a desired output value (also known as human-labeled supervisory signal) train a model. The training data is processed, building a function that maps new data on expected output values. Wikipedia contributors [2024f]. 7, 16

Acronyms

CLB Centre Lèon Bérard. 1, 3, 4

CNN Convolutional Neural Network. 13–15, 26

CNRS Centre National de la Recherche Scientifique. 4, 16

CREATIS Centre de Recherche En Aquisition et Traitement de l'Image pour la Santé. 1, 3, 4

IDRIS Institut du développement et des ressources en informatique scientifique. 1, 16

INSA Institut National des Sciences Apliquées. 4

INSERM Institut National de la Santé et de la Recherche Médicale. 4

PSMA prostate specific membrane antigen. 5

ROI Region of interest. 7–9, 12, 13, 17, 20

References

Anatomical board, None. https://quizlet.com/ca/248964673/anatof-oeil-flash-cards/ Accessed: 2024-06-10.

CLB, None. https://www.centreleonberard.fr/ Accessed: 2024-06-13.

Creatis-Insa-lyon, None. https://www.creatis.insa-lyon.fr/site/en Accessed: 2024-06-12.

Definition CT scan, None. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/computed-tomography-scan Accessed: 2024-06-12.

Definition Deep Learning, None. https://www.ibm.com/topics/deep-learning Accessed: 2024-06-12.

Definition lacrimal glands, None. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/lacrimal-gland Accessed: 2024-06-12.

Definition salivary glands, None. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/salivary-gland Accessed: 2024-06-12.

Yulia Gavrilova. Convolutional neural networks for beginners. *Serokell*, 2021. URL https://serokell.io/blog/introduction-to-convolutional-neural-networks.

Jay Lowe, How to Create Segmentation Masks with Roboflow, Nov 30, 2022. https://blog.roboflow.com/how-to-create-segmentation-masks-with-roboflow/#:~:text=What%20is%20a%20Segmentation%20Mask,in%20a%20computer%20vision%20model. Accessed: 2024-06-14.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. doi:10.1109/5.726791.

Fischer. Olaf, Ronneberger. Philipp and Brox Thomas. U-net: Convolutional networks for biomedical image segmentation. *Space Science Reviews*, 2015. doi:arXiv:1505.04597v1 [cs.CV].

Resample, 2022. https://helpx.adobe.com/lightroom-classic/lightroom-key-concepts/resample.html#:~:text=Changing%20the%20pixel%20dimensions%20of,(center)%20is%20slightly%20blurred. Accessed: 2024-06-14.

Sunaway Cancer Centre, 2023. https://www.sunwaycancercentre.com/en/targeted-radionuclide-therapy/ Accessed: 2024-06-13.

Total Segmentator, 2022. https://github.com/wasserth/TotalSegmentator Accessed: 2024-06-13.

What is data labeling, None. https://aws.amazon.com/what-is/data-labeling/?nc1=h_ls Accessed: 2024-06-14.

What is dice, None. https://towardsdatascience.com/how-accurate-is-image-segmentation-dd448f896388 Accessed: 2024-06-16.

Wikipedia contributors. Hounsfield scale — Wikipedia, the free encyclopedia, 2024a. URL https://en.wikipedia.org/w/index.php?title=Hounsfield_scale&oldid=1222494196. [Online; accessed 15-June-2024].

Wikipedia contributors. Sagittal plane — Wikipedia, the free encyclopedia, 2024b. URL https://en.wikipedia.org/w/index.php?title=Sagittal_plane&oldid=1220910969. [Online; accessed 14-June-2024].

Wikipedia contributors. Cross-validation (statistics) — Wikipedia, the free encyclopedia, 2024c. URL https://en.wikipedia.org/w/index.php?title=Cross-validation_(statistics)&oldid=1227957051. [Online; accessed 16-June-2024].

Wikipedia contributors. Dicom — Wikipedia, the free encyclopedia, 2024d. URL https://en.wikipedia.org/w/index.php?title=DICOM&oldid=1223545023. [Online; accessed 14-June-2024].

Wikipedia contributors. Image segmentation — Wikipedia, the free encyclopedia, 2024e. URL https://en.wikipedia.org/w/index.php?title=Image_segmentation&oldid=1222875710. [Online; accessed 14-June-2024].

Wikipedia contributors. Supervised learning — Wikipedia, the free encyclopedia, 2024f. URL https://en.wikipedia.org/w/index.php?title=Supervised_learning&oldid=1220694111. [Online; accessed 14-June-2024].

Wikipedia contributors. Neuroimaging informatics technology initiative — Wikipedia, the free encyclopedia, 2024g. URL https://en.wikipedia.org/w/index.php?title=Neuroimaging_Informatics_Technology_Initiative&oldid=1220687294. [Online; accessed 15-June-2024].