Few-shots learning of anatomic and oncologic structures in radiology

RAIDIUM Data Challenge Report

Julien AJDENBAUM

julien.ajdenbaum@telecom-paris.fr
MVA, ENS Paris/Saclay
Télécom Paris

17th March 2024

The jury consists of its president : Enzo DALBY and the challenge providers from RAIDIUM : Corentin DANCETTE and Pierre MANCERON

This work is the report for a data challenge[1] provided by RAIDIUM, as part of Stéphane MALLAT's class *"Apprentissage et génération par échantillonnage aléatoire"* at *Collège de France*

Abstract

Abstracts must be able to stand alone and so cannot contain citations to the paper's references, equations, etc. An abstract must consist of a single paragraph and be concise. Because of online formatting, abstracts must appear as plain as possible. Three to six keywords must be included. Each keyword should not exceed three words.

Keywords: Few-shots learning, Self-supervised learning, dice metric, Medical Image Segmentation.

1. Introduction

2. Challenge Presentation

The goal of the challenge is to provide a segmentation of CT scan images. The segmentation provided represents anatomical and oncological structures, but does not cover the

[1] https://challengedata.ens.fr/participants/challenges/150/

entire image. The data provided are only partially annotated: the train set is made up of 2000 images, 400 of which have a segmentation mask. The test consists of 500 images of different structures. Some of such structures are present in the training set (few-shot learning) but some are not (zero-shot learning). The goal of the challenge is to learn this segmentation. As usual in this kind of competition in order to prevent any attempt to overfit, there is public and a private dataset. The accuracy is measured on both sets by averaging the Rand Index between each label and its associated prediction, while also excluding background pixels (i.e. 0 in the label).

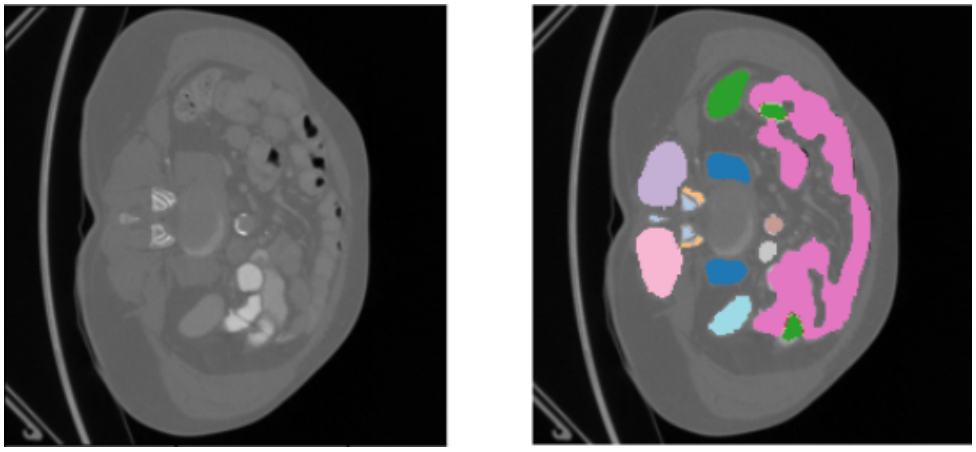

Figure 1: Example of an image and its segmentation

2.1 Understanding the data

The input is comprised of a collection of 2D grayscale images (represented as a 3D numpy array) that depict slices of a CT scan in the transverse plane, each image being of size 512x512 pixels. The order of the slices is random, leading to a lack of 3D spatial context. The output consists of a set of 2D matrices (also a 3D numpy array) with dimensions of 512x512 pixels, containing integer (uint8) values. Each coordinate (w, h) within each matrix $Y_{i,w,h}$ corresponds to a specific anatomical structure.

The segmentation is only partial : the background and unidentified structures are not segmented. The number of segmented regions varies from image to image. Segmented areas are not labeled and are identified with a random integer greater than 1, 0 being reserved for background information.

The number of segmented areas varies greatly from one image to the next, as seen in Figure 2 . There are on average 18 areas per image but with a high standard deviation of 7. The size of each region also varies greatly, with very small regions and much larger regions. One fact that simplifies the task is that every region is continuous and that most but not all regions are separated by a background.

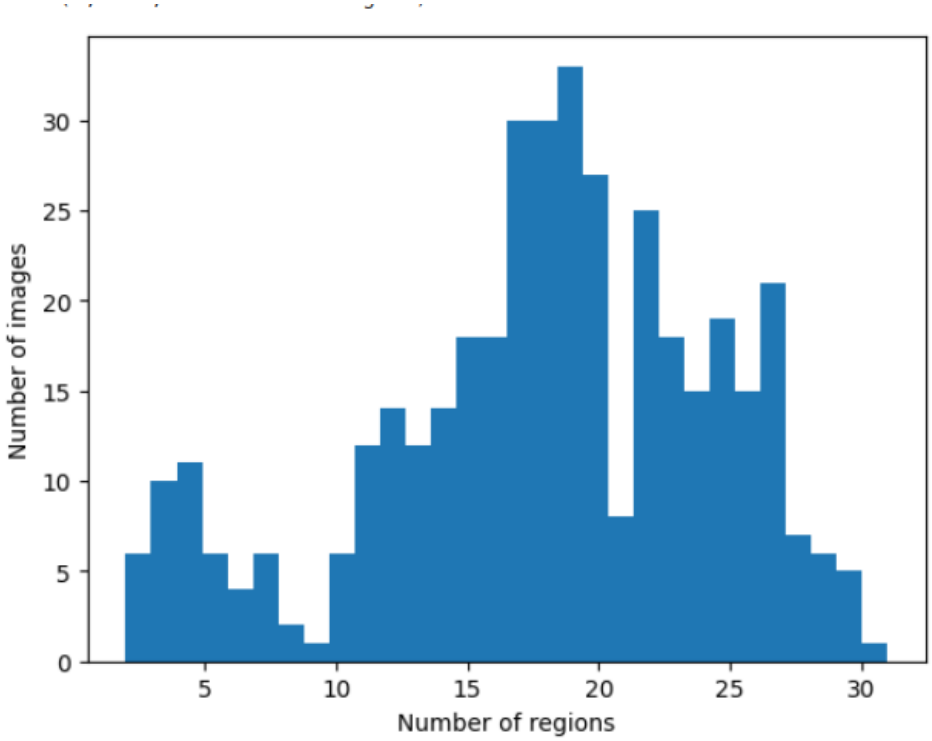

Figure 2: Histogram of the number of segmented regions per image

At first glance, we can guess that some regions will be very easy to segment (very round regions with a high contrast compared to the background) and some others will be much more complex (the small intestine in Figure 1). This is due to the fact that this structure is not homogeneous and non-convex.

Most regions are very small (the average size taken by a region is 1%), but a fifth of the images have at least one region that takes more than 10% of the image (Figure 3). Still in average, 19% of each image is covered by a segmentation mask.

2.2 Accuracy measure

There are many different ways of measuring the performance of segmentation algorithms, each having its specific advantages and purposes.

Pixel Accuracy is the simplest measure, calculating the percentage of pixels in the image that are correctly classified. It is straight forward but can be misleading if the class distribution is unbalanced, as it might favor larger segments.

Intersection over Union (IoU), also known as the Jaccard Index, measures the overlap between the predicted and ground truth segments. It is defined as the size of the intersection divided by the size of the union of the predicted and true segments. This measure is more robust than pixel accuracy, especially in the presence of class imbalance.

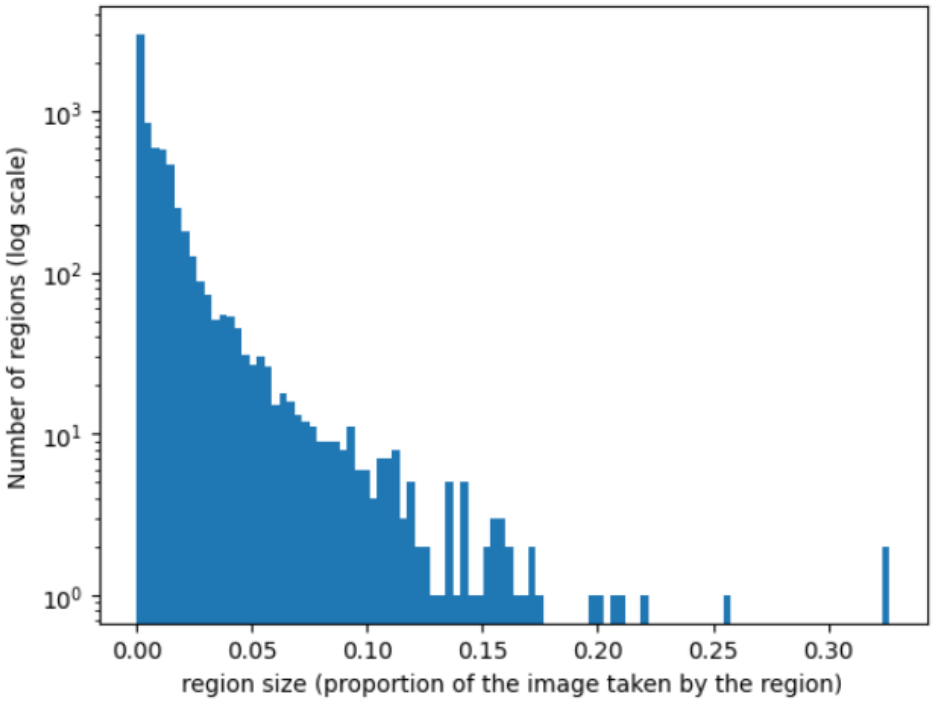

Figure 3: Histogramm of the number of segmented regions per image

Dice Coefficient (F1 Score) measures the overlap but is calculated as the size of the intersection divided by the average size of the predicted and true segments. This is particularly used in medical image segmentation because it balances false positives and false negatives.

Precision and Recall are used when the focus is on a particular segment or class. Precision measures how many of the pixels classified as a specific class are actually that class, while recall measures how many of the actual class pixels were correctly classified. These are important when false positives and false negatives have different costs.

While all of the metrics presented above could be adapted for unlabeled segmentation, matching one area to the other could be a challenge. The measure used in this challenge, rand Index (RI), evaluates the similarity between two data clusterings. Given a set of elements and two partitions of these elements, the Rand Index measures the percentage of agreements (both being in the same segment or in different segments) between the two partitions, ignoring permutations. It's a measure of how similar the clusters are, irrespective of the actual labels.

It values both the true positive and true negative decisions of segmentation, which makes it balanced especially in applications where all types of correct decisions (segmenting together and apart correctly) are equally important. Once again, this makes it particularly suitable for medical applications such as the one we are working with.

The Rand Index metric is calculated as follows :

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

Where RI is the Rand Index metric, TP is the True Positive Rate, TN is the True Negative rate, FP is the False Positive rate and FN is the False Negative rate.

3. Proposed Solutions

3.1 Analysis of the provided baseline

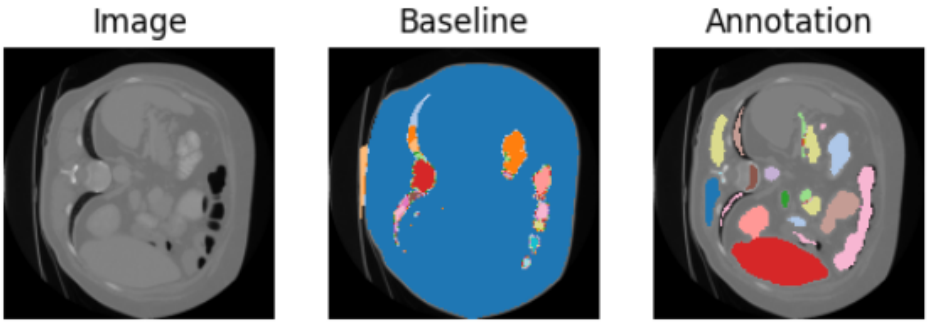

Figure 4: Example of baseline result

The baseline method provided is a widely used baseline for medical image segmentation tasks. It starts with Sobel edge detection to highlight edges. This filter employs horizontal and vertical convolution kernels to compute the image's gradient, highlighting edges by emphasizing regions of high spatial frequency. This step is crucial for identifying the boundaries within the image.
Following this, the image undergoes median filtering, where a median filter is applied over a circular structuring element. This step helps to reduce noise while preserving important edge information, which is essential for accurate segmentation.

After denoising, the method computes the gradient of the denoised image and creates binary markers by applying a threshold. These markers help to distinguish between the objects to be segmented and the background. Subsequently, these markers are labeled to identify distinct objects in preparation for segmentation.

Finally, the watershed segmentation algorithm is applied. This technique uses the previously detected edges and the labeled markers to segment the image. The compactness parameter in this step controls the shape of the regions, influencing how the segmentation

adapts to the image's features.

The baseline is very simple, fits in a few lines of code, and has a not too shaby result (metric : 0.14). However, there are several drawbacks. The method can lead to over-segmentation, dividing single objects into multiple segments. It's also sensitive to hyperparameter settings, which might require manual adjustment for different datasets. But of course the main drawback is that it there is no learning. It only works with simple operations on the image, but it does not necessarly match with what experts are interested in. Without this crucial information it makes many errors. For example, the smaller regions of the body, which are considered as background in the annotation, is considered as a region in the baseline.

3.2 General Purpose Pretrained Model

The second baseline we can take is the result of a general purpose pretrainted model. In our case, we tried the Segment Anything Model (SAM) Kirillov et al., 2023. SAM is a novel AI approach from Meta AI designed for image segmentation. It's a promptable segmentation system that showcases zero-shot generalization capabilities, meaning it can segment objects in images it has never seen before without additional training. SAM stands out because of its ability to accurately "cut out" or segment any object within any image with just a single click, based on the prompts given. This makes it highly versatile and powerful for various image segmentation tasks.

We use

3.3 Self-Supervised Learning : a pretraining of the network

3.4 Learning the segmentation masks

4.

The entire project was run on Télécom Paris' GPU servers. Those machines are made for research by both students and PhD students, and due to low usage (many machines were unused), we ran it on an overkill machine. The machine is comprised of 3 A-100 Nvidia GPU's with 40Gb of RAM each, 48 Intel Xeon Gold CPUs and 400Gb of system RAM. The machine is controlled via SSH and using a remote jupyter notebook.

References

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment anything.