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

18th 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

The objective of this challenge is to segment anatomical and oncological structures in 2D Computed Tomography (CT) images, utilizing a dataset that includes both fully annotated and unannotated images. The dataset includes a large amount of medical images, which are used for segmentation tasks where the segmented structures do not encompass the entire image, leaving certain pixels unaccounted for and considered as part of the background. In this report, we show the different techniques tried to tackle this problem. We show that the tested SSL pretraining techniques, while very promising in some cases, are not well adapted to our case since the amount of data available is not big enough. A very simple region-based class separation leads to good results on the validation dataset of 86% for a binary classification (all the classes are assimilated to one class). The Adaptive t-vMF result proved to be a good regularized method.

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

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

1. Introduction

The objective of this challenge is to segment anatomical and oncological structures in Computed Tomography (CT) images, utilizing a dataset that includes both fully annotated and unannotated images.

The dataset provided comprises 2000 CT images, differentiated into two main categories: 400 images accompanied by detailed segmentation masks of individual structures, serving as the ground truth for training, and 1600 raw, unsegmented CT images that can be utilized in an unsupervised learning context.

The test set includes 500 additional images with corresponding segmentation maps, focusing on both anatomical structures and tumors. The nature of the dataset presents a unique combination of zero-shot and few-shot learning scenarios due to the incomplete overlap of structures between the training and test sets and the limited examples of some structures.

The principal challenge lies in accurately segmenting and identifying diverse anatomical structures within the CT images, where the segmented structures do not encompass the entire image, leaving certain pixels unaccounted for and considered as part of the background. This task is particularly challenging as it involves dealing with variations in structure appearance, size, and shape across different patients and images.

In this report, we show the different techniques tried to tackle this problem. First, we tried self- or weakly-supervised techniques. Those techniques directly came to mind due to the nature of the problem annotations (20%), but came with its set of challenges. Then we tried a Dice Loss based method. In conclusion, we show what techniques we would use if the project were to be done again.

The entire project was run on Télécom Paris' GPU servers. Those machines are made for research by both master's 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.

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 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).

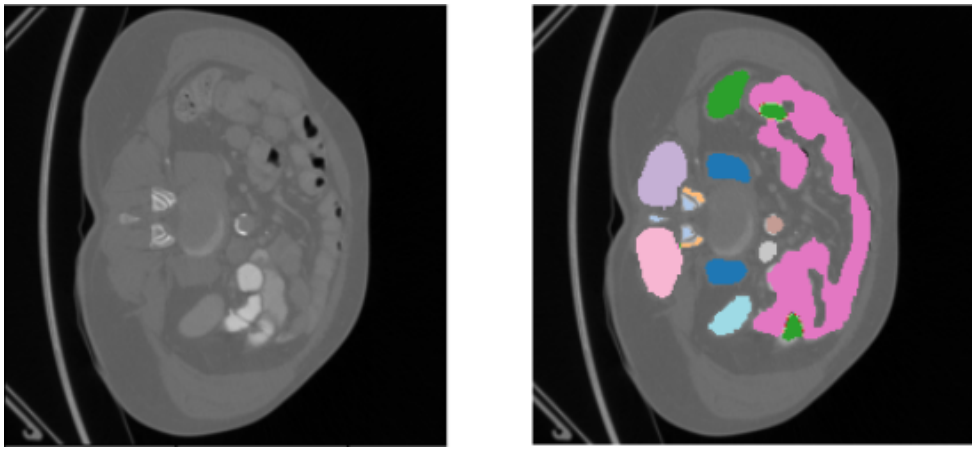

Figure 1: Example of an image and its segmentation.

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 Yi,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.

At first glance, we can guess that some regions will be very easy to segment (very

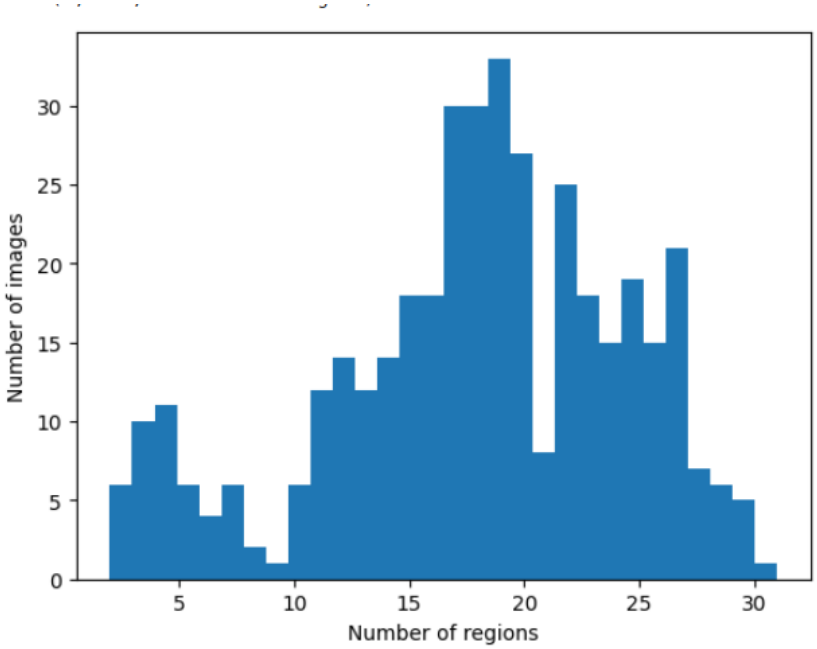

Figure 2: Histogram of the number of segmented regions per image.

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.

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 par-

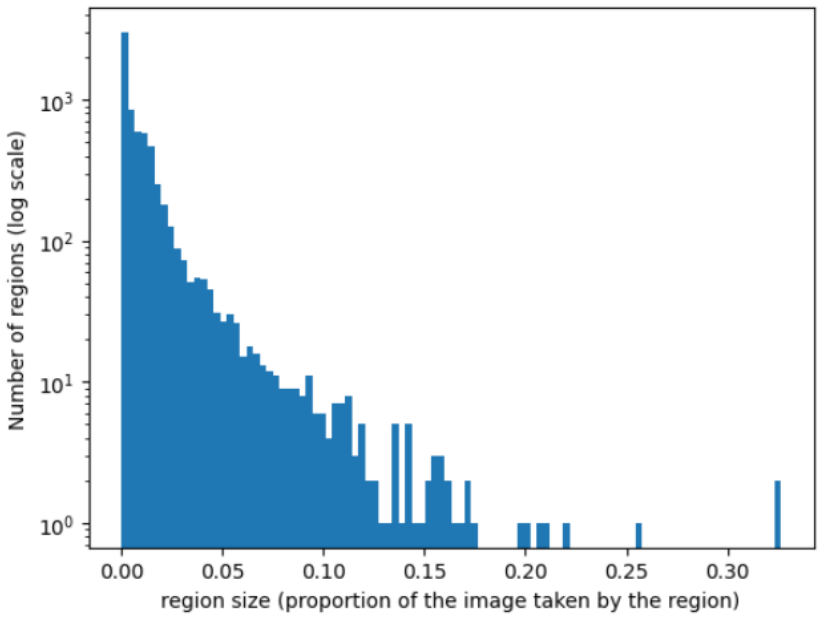

Figure 3: Histogram of the size of segmented regions per image.

ticularly 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

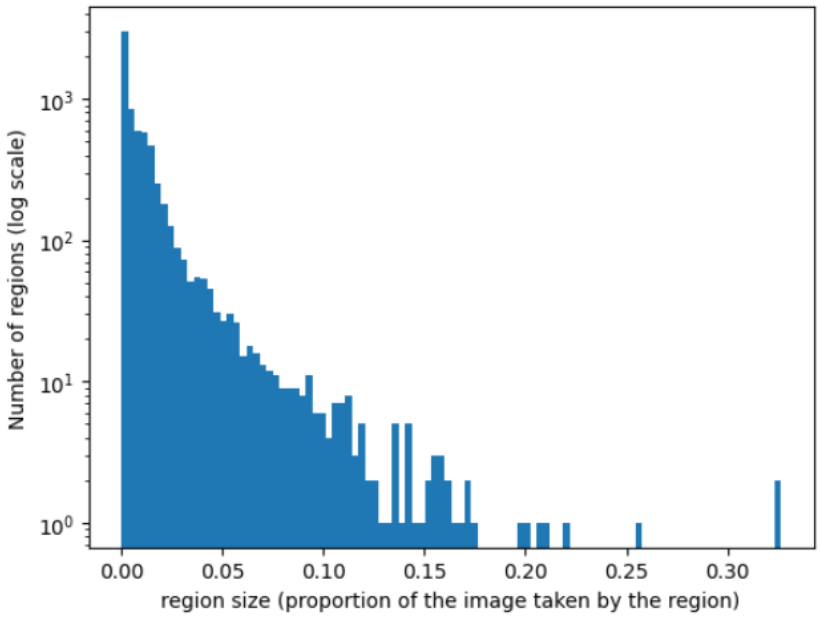

Figure 4: Example of baseline result.

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.

SAM was trained using a large dataset comprising 11 million images and over 1 billion masks. SAM consists of three main components: an image encoder, a prompt encoder, and a mask decoder. The image encoder processes input images and produces a single image embedding. The prompt encoder is designed for efficient encoding of different prompt modes, enabling the model to understand and respond to various segmentation tasks based on textual prompts. Lastly, the mask decoder combines the image embedding 148 with the prompt encodings to generate segmentation masks quickly and accurately.

The training process allowed SAM to learn from a vast amount of visual information and corresponding segmentation masks, enabling it to understand complex relationships between images and segmentation tasks. As a result, SAM demonstrates zero-shot transfer abilities, which means that it can perform segmentation on new images and tasks it has not explicitly seen during training. This broad capability is due to the comprehensive nature 155 of its training dataset and the effectiveness of its architectural design.

We use the prompt-free version of the model. We obtain 0.14 on the RI metric, the

same as the baseline. This shows something very important : the difficulty here is not to segment the different areas of the image (in contrary to hair in a natural image for example, which is very hard to segment). The difficulty here is to understand the interests of doctors. And this information can only come from learning the dataset.

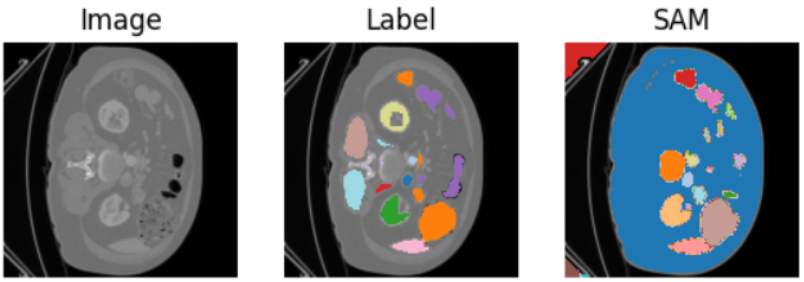

Figure 5: SAM Results.

3.3 Self-Supervised Learning : a pretraining of the network

In this problem, we only have 20% of annotations. This means that to make the best use of all the available information, any algorithm should try to learn from unsegmented images. Hence, the use of Self-Supervised Learning (SSL).

Such SSL methods have become increasingly prevalent, particularly for segmentation tasks where annotated datasets are limited and very hard to build. SSL methods exploit unlabeled data to learn meaningful representations that are beneficial for downstream tasks such as segmentation. A key component of SSL is the use of pretext tasks, which are learning objectives designed to train a model without the need for annotated examples. These tasks are constructed to encourage the model to understand the inherent structures and patterns in the data that are relevant for the main task but do not require manually labeled data.

At the heart of SSL, pretext tasks are engineered challenges that a model must solve, thereby learning useful features from the data. These tasks are created from the data itself, without relying on external annotations. The design of a pretext task is critical, as it guides the model to discover and leverage the intrinsic properties of the data relevant to the medical segmentation challenges. The effectiveness of an SSL approach largely depends on the relevance of the chosen pretext tasks to the primary task of interest.

If we had access to more specific informations about the data, such as the entirety of the slices for a patient or each slice number, we could use slice number prediction as a pretext task. In such architectures, the model is trained to predict the relative or absolute position of a given slice within the entire volume. This task does not require any external labels, as the slice position serves as a self-generated label based on the slice's order in

the scan. By predicting the position of a slice, the model learns to capture anatomical features and variations along the axis of the scan, which are crucial for understanding the spatial relationships and structures within the body.

3.4 SimCLR

An example of a pretext task which we can use is contrastive learning, an SSL approach where the model learns by trying to find, in a batch of images, the pairs that originally are the same and went trough two different transformations. This serves as a pre-training before the downstream task of segmentation. The pre-training has nothing to do with the main task but helps with learning the data distributions and good representations of the images. This idea was introduced with the SimCLR paper Chen et al., 2020, which we will use.

We first take a mini-batch of images, 64 in our case. And we augment each image through two different augmentations, leading to 128 images. As augmentations, we tried a mix of random cropping, random Gaussian noise, random Gaussian blur. A pair of images that are two different augmentations of the same image are called positive pairs and are called negative pairs if they come from two different images. The goal is to ensure that the representation learned by the network is close for positive pairs and lower for negative pairs.

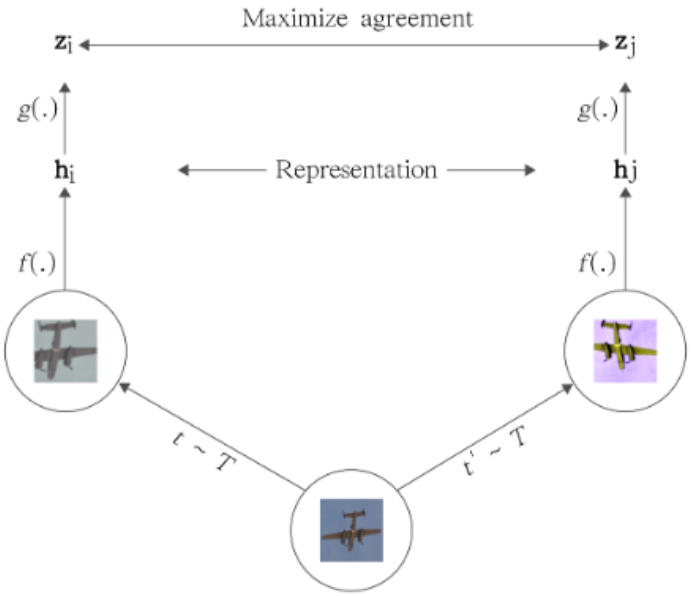

Figure 6: SimCLR Results.

In practice, we want to train a large base encoder, in our case ResNet-50. To adapt it to the specific SimCLR task, we remove the last layer and add a small projection head. We use the NT-Xent loss to measure the similarity between representations :

$$\ell(i,j) = -\log \frac{\exp\left(s_{i,j}/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp\left(s_{i,k}/\tau\right)} \tag{2}$$

This allows pre-training of the ResNet-50 encoder. If this had worked, we would have been able to use this as an encoder and then add a decoder, a class-equivariant loss and train this Auto-Encoder on the labeled dataset.
Unfortunatly, SimCLR requires a lot of images. The number of images we had was not sufficient to appropriately train the network without just overfitting. On the validation dataset, we did not obtain a much better contrastive result than pure randomness.

Even if it had worked, we would have pretrained without supervision the Encoder, but not the decoder. Having to learn the decoder only on labeled data would of course have been a huge downside to this method.

3.5 Swin MAE

The second idea which I kept is image reconstruction, where models are tasked with reconstructing images from their corrupted versions. This approach, forces the model to identify and learn the essential anatomical features while discarding irrelevant noise or artifacts. Through this process, the model gains an understanding of the underlying structures within the medical images, which are crucial for accurate segmentation.

In my case, I used masking as a pretext task, in a method called MAE He et al., 2021. This method consists in training an autoencoder to reconstruct images that are partially masked.

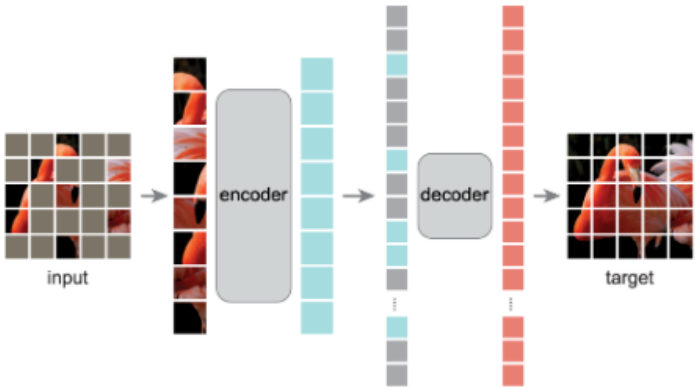

Figure 7: MAE Architecture.

Therefore, we mask the image, and just train it to try and "guess" or reconstruct

the areas hidden under the masks. This method is very interesting since it allows us to pre-train an entire auto-encoder, which can then much more easily be adapted for the segmentation task.

Unfortunately, once again we do not have enough images to correctly train the MAE 8. We tried Swin MAE Xu et al., 2023, a special version made for small datasets, but even then the results were not good enough. The paper tries its method on at least 10k color images, much more than our 4k images.

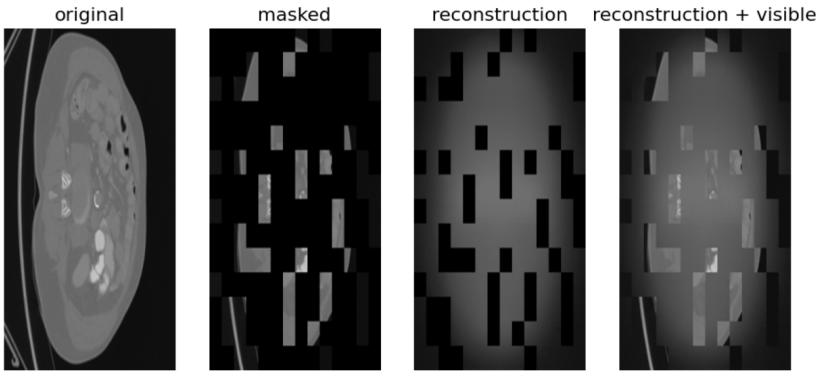

Figure 8: Example of Swin MAE results.

3.6 Conclusion on SSL pretraining techniques

We conclude that the tested SSL pretraining techniques, while very promising in some cases, are not well adapted to our case since the amount of data available is not big enough.

We could also have tested methods similar to other big Foundation Models for images, such as the one proposed by Fair, DINOv2 Oquab et al., 2024. The amazing thing about those models is that they can be easily adapted to do many different tasks (classification, regression, segmentation, depth estimation, ...). The problem is that it requires a lot of data. Estimates are that a good DINOv2 for a foundation model in radiology requires at least between 50k and 100K images.

3.7 Dice loss based methods

Due to the unsuccessful nature of my trials on SSL methods, I tried to return to the basics. Once again, finding the structures and segmenting them did not seem like the most difficult part of the challenge, but understanding which parts are important and seperating them was. Hence, I decided to focus on "simpler" methods.

This return to the basics came in the form of the very basic segmentation loss, very widely used especially in medicine, the Dice loss:

$$\text{Dice loss } = \frac{1}{C}\sum_{i=1}^{C}\left(1 - \frac{2\sum_n A_{in}B_{in} + \gamma}{\sum_n A_{in}^2 + \sum_n B_{in}^2 + \gamma}\right) \tag{3}$$

Where C indicates the number of classes, n indicates the number of class samples, A_{in} indicates the vectors that contain all positive examples predicted by a specific model, and B_{in} indicates the vectors that contain all positive examples of the ground truth in the data set.

To make it more adaptable, an Adaptive t-vMF Dice Loss has been proposed Kato & Hotta, 2022.

We can then normalize each class' segmentation so that $\sqrt{\sum_n A_{in}^2} = \sqrt{\sum_n B_{in}^2} = 1$ and use cosine similarity to obtain this new formulation of the dice loss :

$$\text{Dice loss }_{\text{norm}} = \frac{1}{C}\sum_{i=1}^{C}(1 - \cos\theta_i) \tag{4}$$

Where $\cos\theta_i = \sum_n A_{in}B_{in}$.

Finally, we extend the cosine similarity by using the t-vMF similarity :

$$\phi_t(\cos\theta;\kappa) = \frac{1+\cos\theta}{1+\kappa(1-\cos\theta)} - 1 \tag{5}$$

The parameter κ allows us to have more flexibility in the compactness of the similarity. If $\kappa = 0$, we have the original cosine similarity. Including this t-vMF, we obtain the following final t-vMF Dice Loss :

$$t\text{-vMF Dice loss } = \frac{1}{C}\sum_{i=1}^{C}\left(1 - \phi_t\left(\cos\theta_i;\kappa\right)\right)^2 \tag{6}$$

Like in the original paper, we trained a U-net from Shibuya & Hotta, 2020 it on 4000 iterations, with a train-test data split of 80%. A very simple region-based class separation leads to good results on the validation dataset of 86% for a binary classification (all the classes are assimilated to one class). The test set is comprised of few-shots and zero-shots tasks (some structures are observed in the training set and some are not). This lowers the score for the test dataset. The final public test score was 76.19% and private test score was 76.16%

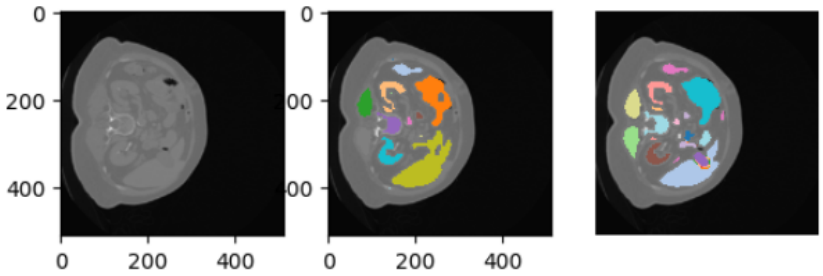

Figure 9: Example of Adaptive t-vMF Dice Loss result.

The benefits of this method is that it is very simple and, therefore, very robust. This is especially useful for zero-shot applications. The main problem that affects our results is that some classes are adjacent. In our method, those classes are merged and, therefore, not taken separately. This is the main point if we are searching for improvements : find a way to make it multi class but permutation invariant Asai, 2018. Lots of hyper parameter and model fine-tuning is also left to be done

4. Conclusion

Throught this challenge we tested numerous different methods. Self-Supervised and Weakly-Supervised methods did not yield good results due to the limited number of training images (2000). The Adaptive t-vMF result proved to be a good regularized method.

If this challenge were to be done again, I would focus on class-agnostic specific methods. I believe that a simple CNN AE but with a good loss that is permutation-equivariant and yet differentiable would do the job perfectly. This AE could first be trained to reconstruct the images. Then the taks could be changed to class-agnostic segmentation loss minimization. Such a method would probably lead to very respectable results.

References

Asai, M. (2018). Set cross entropy: Likelihood-based permutation invariant loss function for probability distributions.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2021). Masked autoencoders are scalable vision learners.

Kato, S., & Hotta, K. (2022). Adaptive t-vmf dice loss for multi-class medical image segmentation.

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.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., ... Bojanowski, P. (2024). Dinov2: Learning robust visual features without supervision.

Shibuya, E., & Hotta, K. (2020). Feedback u-net for cell image segmentation.

Xu, Z., Dai, Y., Liu, F., Chen, W., Liu, Y., Shi, L., Liu, S., & Zhou, Y. (2023). Swin mae: Masked autoencoders for small datasets.