ENS x Raidium Challenge - Few-Shot Learning

Sarra Ben Yahia and José Ángel García Sánchez

Abstract

This paper describes our work on the ENS x Raidium challenge that aims to automatically segment the anatomical structures of the human body from CT scans without semantically identifying associated organs. The challenge focuses on identifying visible shapes on a CT scan, without exhaustive annotations. While supervised segmentation algorithms for individual structures are considered resolved, generalizing to new anatomical structures not seen before is not possible with supervised learning.

Keywords: Deep Learning, SAM, MedSAM, Instance Segmentation, Few Shot Learning, U-net, Out-Of-Domain Learning

1 Introduction

1.1 Problem presentation

The project at hand involves developing a solution for segmenting anatomical structures from CT scans using few-shot learning and instance segmentation techniques. Few-shot learning involves learning from a small number of examples, allowing for generalization to new, previously unseen examples. The challenge in this project is to generalize to new anatomical structures not seen before, which presents an out-of-domain learning problem. The objective is to create a robust algorithm that can accurately segment anatomical structures without semantically identifying associated organs, which is a critical need in medical imaging.

1.2 Database presentation

The training data consists of two types of images:

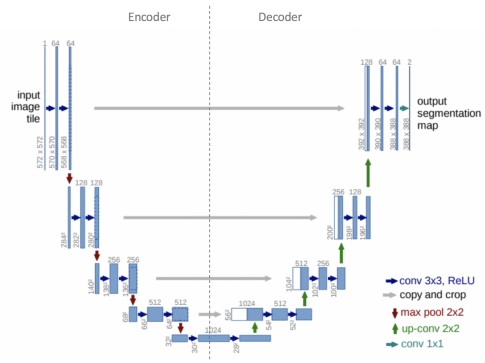
- CT scans with anatomical segmentation masks of individual structures and without.
- CT scans without any segmented structures, requiring a combination of supervised and unsupervised learning approaches.

The test set measures the ability to correctly segment and separate the different structures on an image, with non-identifiable pixels considered as part of the background. However, as the data was part of a challenge, the test-set wasn't directly available to us. To see our results, we had to submit some prediction in a strict format to have our performance computed in Rand index. Therefore, we will see why this was a challenge for us.

2 Presentation of the models implemented

2.1 U-net

The U-Net is a convolutional neural network architecture designed for image segmentation tasks. It was first introduced by Ronneberger et al. in their 2015 paper "U-Net: Convolutional Networks for Biomedical Image Segmentation". It has an encoder path that extracts features from the input image through convolutional and max pooling layers.
The output is a feature map containing high-level representations of the image. The decoder path is a mirror of the encoder path and upsamples the feature map to produce a segmentation mask with the same size as the input image.
The central bottleneck reduces the size of the feature map produced by the encoder path and captures the most important features of the input image. The U-Net is widely used in medical image analysis due to its ability to accurately segment biomedical images and that is why it was our first choice.



Mathematically, the U-Net architecture can be represented as a function that maps an input image x to a segmentation mask y:

$$y = f(x) \tag{1}$$

where y is the segmentation mask, x is the input image, and f is a complex non-linear function that is parameterized by the weights of the convolutional and deconvolutional layers. During training, the U-Net architecture is optimized to minimize a loss function that measures the discrepancy between the predicted segmentation mask and the ground truth mask:

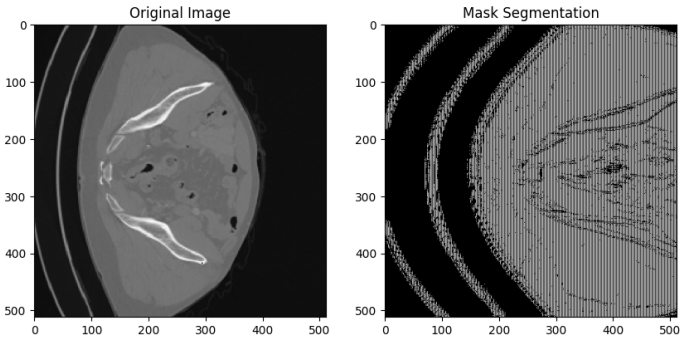
$$L = -\sum_i y_i \log(f(x)_i) + (1 - y_i) \log(1 - f(x)_i) \tag{2}$$

where y_i is the binary label for pixel i and $f(x)_i$ is the predicted probability of pixel i belonging to the foreground class.

In summary, the U-Net architecture is a powerful tool for image segmentation tasks. The U-Net model we trained from scratch was only trained on the available 200 labeled observations, which may not be sufficient to achieve high accuracy. Furthermore, due to the limited computational power of our devices, we were not able to train the model with a large number of epochs and a large batch size. Even using Google Drive, we were very limited in terms of memory and couldn't train the model properly.

Since the U-net model requires a specific number of targets for segmentation, we made a decision to merge all classes of target into a single class. Our objective was to identify organs rather than annotate the segmentation. Therefore, we assumed that providing the overall masks as ground truth would suffice to generalize the model to new anatomical structures.
It is important to note that the decision to merge multiple classes of targets into one can have advantages and disadvantages.
On one hand, this can simplify the annotation process for annotators, thereby reducing the time and costs associated with annotation. Additionally, this can help generalize the model to other anatomical structures by learning common features.
On the other hand, this can also introduce classification errors by removing subtle differences between the different classes of targets. This can affect the accuracy and quality of segmentation.

Result of the U-net on the test set :



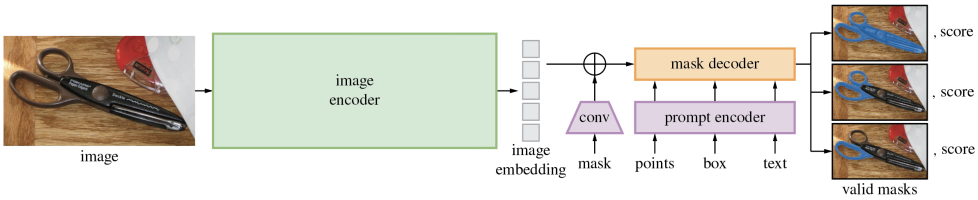
Unfortunately, as we do not possess the ground truths for our test set, we are unable to calculate performance metrics based on our predictions. Nevertheless, the visual representations that we have observed do not appear to be very encouraging anyways.

We considered using pre-trained Resnet models as encoders to our U-net to utilize their weights and thus address our lack of data. However, the major problem was that our CT scans were grayscale images and did not have an RGB format, on which Resnet is trained and therefore expects as input. We made some attempts to format them, but upon more research, we concluded that such models, being supervised, did not correspond to our problem of instance segmentation without organ identification. Therefore, we decided to turn to other types of models.

2.2 Segment Anything Model (SAM)

The Segment Anything Model (SAM) is a model developed by the Meta AI Research Team and is available in Open Source. It is used mainly for image segmentation tasks. It is a model that takes an input image and produces a pixel- wise classification map for the various objects present in the image. The model architecture is designed to segment images into arbitrary objects with or without semantic labels. It was the perfect model for our problematic.

Let's dive into the details of the SAM architecture: The researchers used a pre-trained



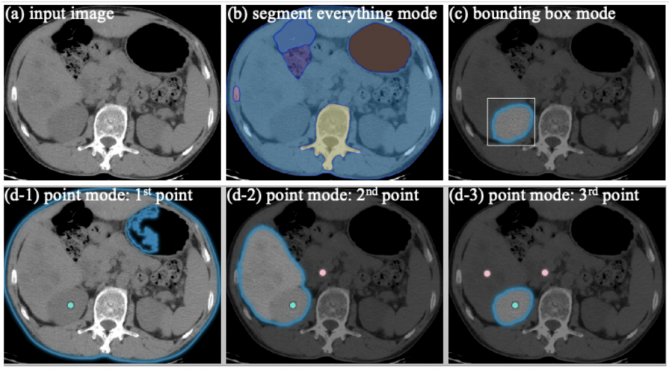
Vision Transformer (ViT) that was trained with Mean Absolute Error (MAE) to encode images. The ViT was adapted to handle high-resolution inputs and runs once per image. It can be used before prompting the model. Two types of prompts were considered: sparse (points, boxes, text) and dense (masks). Points and boxes were represented by positional encodings, while free-form text used an off-the-shelf text encoder from CLIP. Dense prompts were embedded with convolutions and added to the image embedding.

The mask decoder efficiently maps the image embedding, prompt embeddings, and an output token to a mask. This design, inspired by previous work, employs a modification of a Transformer decoder block followed by a dynamic mask prediction head. The modified decoder block uses prompt self-attention and cross-attention in two directions (prompt-to-image embedding and vice-versa) to update all embeddings. After running two blocks, the image embedding is upsampled, and an MLP maps the output token to a dynamic linear classifier. The classifier then computes the mask foreground probability at each image location.

To prevent ambiguity, the model predicts multiple output masks for a single prompt. They found that three mask outputs were sufficient to address most common cases. During training, they backprop only the minimum loss over masks. To rank masks, the model predicts a confidence score (i.e., estimated Intersection over Union) for each mask.

The model was supervised with a combination of focal loss and dice loss and trained for the promptable segmentation task using geometric prompts. To simulate an interactive setup, they randomly sampled prompts in 11 rounds per mask, allowing the model to integrate seamlessly into their data engine.

We used the "segment everything mode" of the model, because we considered it was the most adapted for our problematic. For illustration, here is a graphic of segmentation results of SAM based on different segmentation modes.



- Results with SAM on our data

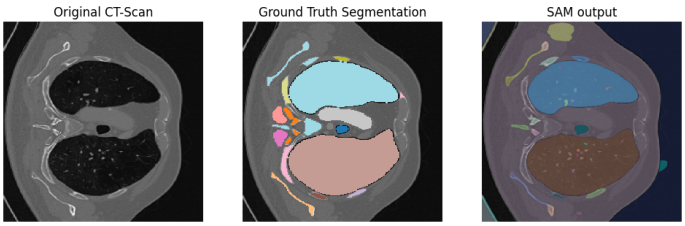


Fig. 1 Comparison between SAM output and ground truth

We can observe that SAM tends to the target parts of the CT-SCAN rather accurately. However, a major concern here is that it tends to segment everything (as the mode tells us, no surprises here). We expect then that the segmentation of the background and the main anatomy of the human body will decrease the overall metrics.

- Computed metrics

Table 1 SAM Performance Metrics

Metric	Value
Average IOU	0.0276
Average DSC	0.0430
Average Precision	0.4272
Average Recall	0.0439
Average Rand Index	0.0067

The segmentation results obtained using the "segment everything" mode of SAM were found to be suboptimal, as indicated by the low IOU, DSC, and Rand Index scores. Specifically, the average IOU and DSC scores were found to be 0.0276 and 0.0430, respectively, suggesting a limited overlap between the ground truth and predicted segmentations. Similarly, the Rand Index score was very low at 0.0067, indicating a performance no better than random segmentation.

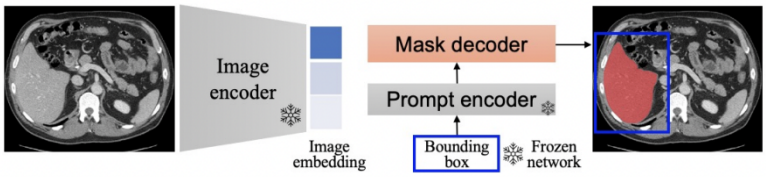
However, the average precision score was relatively high at 0.4272, indicating that the model correctly identified 42 percent of the pixels belonging to the segmentation. In contrast, the recall score was found to be very low at 0.0439, suggesting that the model only identified a small proportion of actual pixels that belonged to the segmentation.

One potential explanation for these sub-optimal results may be due to the "segment everything" mode of SAM, which attempted to segment all structures in the CT scans.

These results are expected. Moreover, the use of SAM in medical image segmentation is limited due to the considerable differences between natural and medical images.
Moreover, our personal problematic is different and we see that comparing the ground truth with SAM output seems silly.
Our focus will be on the precision metric of the next model, as it measures the accuracy of the model in identifying the ground truth. Since SAM operates on this principle, the other metrics are expected to be relatively low.

2.3 Med-SAM

Med-SAM is a deep learning architecture specifically designed for medical language processing tasks.It is basically a fine-tuned version of SAM on medical images. For



the fine-tuning process, the authors chose to freeze all the components of the original SAM, meaning that they did not modify or update the existing parameters of the model's backbone network during training. Instead, they only fine-tuned the mask-decoder, which is responsible for generating object masks from the features extracted by the backbone network.

Initially, MedSAM seemed suitable for our problem. As it was fine-tuned on CT scans, we thought that this fine-tuning could help SAM better segment our objects. We focused on the pre-processing of the data, which was in 3D, while ours was in 2D. This is one of the most complicated steps. We had to delve into the original code in detail and try to adapt the 3D image formats to the 2D pngs we had.

However, this allowed us to truly delve into the code in detail. Our images and labels were in grayscale, with 104 different labels. Since MedSAM used binary labels, we converted our grayscale labels to binary. This resulted in multiple labels for a single image; for instance, if we initially had an array with grayscale values, we now have binary arrays equal to the number of unique grayscale values in the image. All of this is stored in NPZ format, with a shape like (512x512x19), assuming there are 19 labels in the image. The NPZ format is a compressed numpy array containing both the images and labels. However, we then realized that MedSAM only accepted bounding boxes that couldn't predict multiple labels simultaneously, while our requirement was to segment the entire image into different objects.

We think the right approach for the future will be to add the possibility to use SAM's segment everything mode in the MedSAM code and fine-tune it with the 1TB of data used for MedSAM, in addition to our data, to predict on our test dataset.
At our current stage and due to lack of time (and GPU), we couldn't explore this path further, but we will make sure to continue with the project.
Note: The segment everything mode allows for multiple ground-truths in the NPZ, which would help us address our problem.

3 Contributions to Github repositories

The project presented a significant challenge, particularly when it came to handling the large dataset and identifying multiple segmentation's on the same image. To address this issue, we realized that we needed a pre-trained model with a robust dataset that was adapted to the problematic.

We initially attempted to adapt the source codes of Med-SAM and SAM, but encountered numerous errors that required adjustment to make the models suitable for our purposes. Consequently, we contributed to the Med-SAM repository by correcting the code errors and merging to main original branch.

Moreover, while the SAM model was effective in distinguishing different parts, it also segmented the background and larger structures and it decreased immensely to the metrics against ground truths. To address this issue, we explored the "min mask

region area" option that allowed us to set a minimum size for the segment. However, our situation required a "max mask region area" option, which unfortunately we could not find in the existing model.

To address this issue, we submitted a pull request on the facebookresearch/segmentanything repository to ensure that the model was customized for our specific needs. We are still waiting for the pull request to be accepted.

4 Areas for improvement

Besides the improvements we discussed earlier, here are some other ideas that could help us find effective solutions.

1) Obtaining additional labeled data to augment our training set would be beneficial. It would have been interesting to see how fine-tuning SAM or even Med-Sam on our particular data to make it more specific to its unique problematic would benefit the performances. However, in the context of our project, fine-tuning Med-Sam may not be a viable solution since it could potentially lead to higher variance and reduced model generalization.

2) The next method that we could consider would be the following: Use Med-Sam (which is more accurate in its predictions) in "bounding box" mode and regrouping the predictions. The method would be as follows:

- For the same image, make predictions several times on different tasks. For example, for a CT-scan in the coronal plane, one would launch a prediction for the task of artery segmentation, another for the colon, etc...
- At the end of these multiple prediction tasks on the same CT-scan, all segmented "masks" are grouped together.
  This way, it would allow us to use Med-Sam's most efficient mode and at the same time be able to annotate our segments.

4) Using other algorithms: One could use the BAM model (https://paperswithcode.com/method/bam), but it is necessary to change the training annotations and leave only one target per image (the same image can be repeated several times).

5 Conclusion

In conclusion, this is a project that we really enjoyed, and we plan to continue working on it to improve our skills.

Overall, our journey has been challenging, but we have made progress towards our goal of identifying multiple segmentation on the same image. Besides, contributing to large open source projects was very engaging and let us explore a whole other dimension of the open-source movement.
We will continue to work on this until the end of the challenge, and hopefully discover and help to develop a adapted solution to the problem.