ADL - Assignment 2

Julian Barragan Gutierrez - rbs813

June 14, 2024

Contents

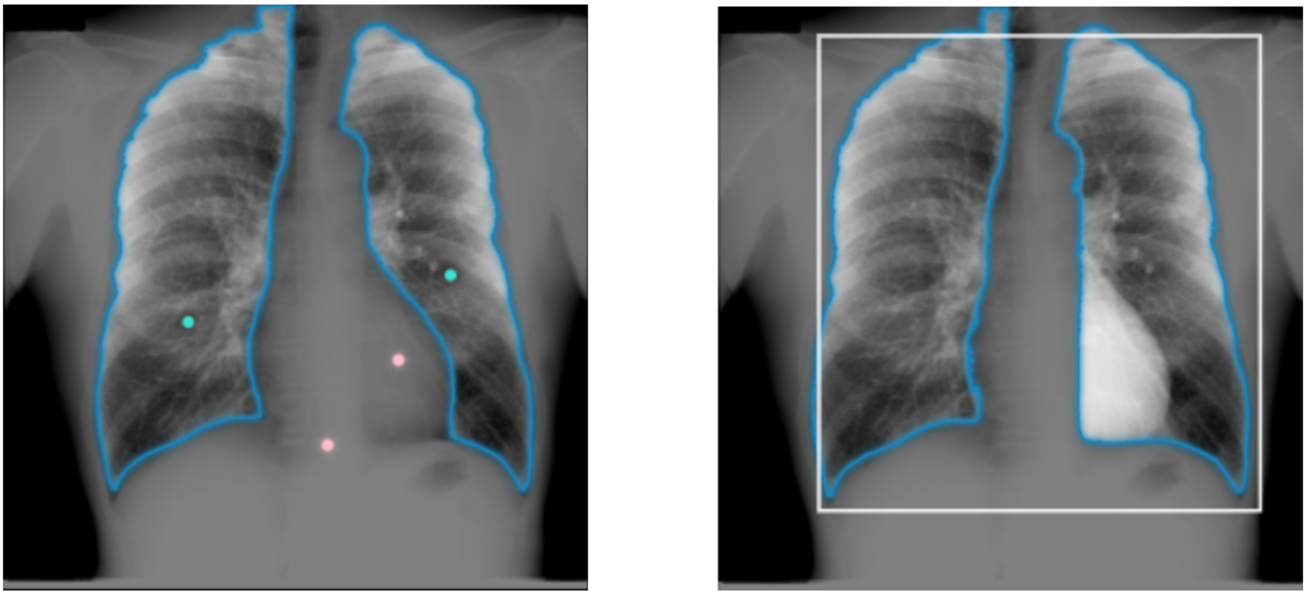
1 Segment Anything

1.1 Brief discussion

Based on the paper, the model exhibits strong zero-shot segmentation capabilities. The model is trained on a (very) large and diverse dataset, and shows some generalisation unto unseen data. However, medical image segmentation of x-ray requires high precision. The model has been tested on the *PIDRay* dataset of x-ray images, indicating some level of generalizability.

In conclusion the model may exhibit some zero-shot transfer capabilities, but with no specific training on medical x-ray image segmentation we don't expect high precision.

1.2 Brief analysis



(a) Click prompt

(b) Box prompt

Figure 1: Click and Box prompt of test image

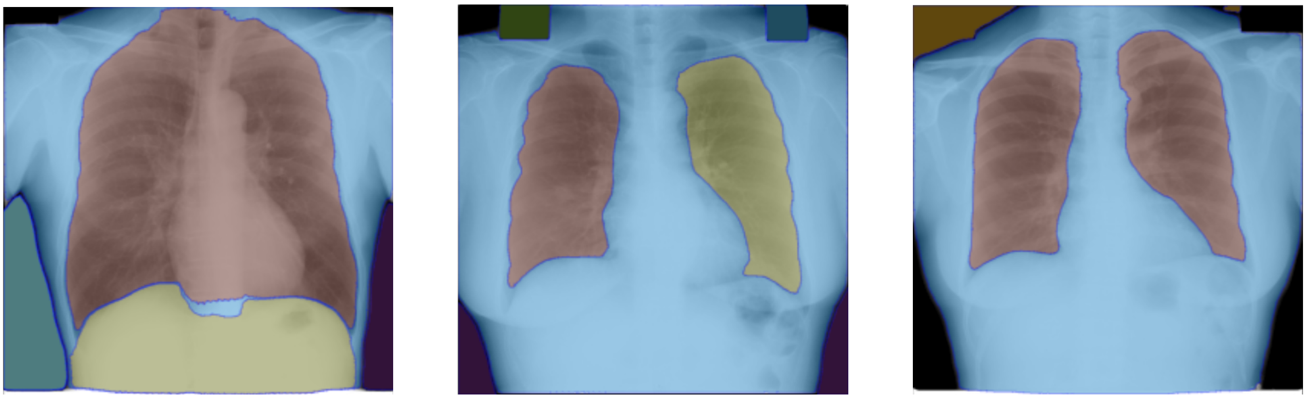
Figure 1 shows the prompting methods *click prompt* and *box prompt* on the first or three test images. The advantage of *click prompting* is, that it provides the ability to deselect area which aren't to be included in the segmentation mask(s).

Comparing the two segmentations in figure 1 we see that the *box prompt* provides decent masks, but the right mask includes an undesirably portion next to the left lung, which doesn't seem to be part of the lung. The *box prompt* method is fast, easy to use and powerful, but we are limited to segmenting within the rectangular confines of the box.

However with the *click prompt* we are able to deselect undesired areas (red dots). Indeed, with just the green dots the segmentation was very similar to the box prompt, but the inclusion of red dots allow us be more precise.

We seem to be able to get higher precision by manually clicking rather than using *box prompting.*

While it is possible to engineer the click prompts in an image, for automation, the *click prompting* precise coordinate selection is a much more demanding task than determining a region of an image for *box prompting.* In further sections we will experiment with prompt engineering.



(a) First image

(b) Second image

(c) Thirds image

Figure 2: Segment everything prompt on all three images

In figure 2 we see the *segment everything* prompting method, which returns all the segmentation which SAM internally evaluates as the most relevant.

While this indeed provides strong zero-shot segmentation capabilities, figure 2 clearly shows the ambiguity in the prompts. SAM is not trained on this specific segmentation tasks, thus it will ambigously choose the most relevant segmentation. The second and third image provide a very good segmentation of the lungs in fact, but in the second image (b) SAM makes two masks; one for each lung. In the third image (c), SAM provides a single mask for both lungs. In the first image (a), the segmentation masks are useless w.r.t. lung segmentation.

Thus we see that *segment everything* provide zero-shot segmentation capabilities, but has no conception as to what the user desires from the segmentation, and just autonomously decides which masks to output. With *click prompt* and *box prompt* we have higher influence in the segmentation, and while *click prompt* provides the highest precision potential, it also demands the most interaction with the model.

2 Prompt Engineering in Python

2.1 Prompting SAM

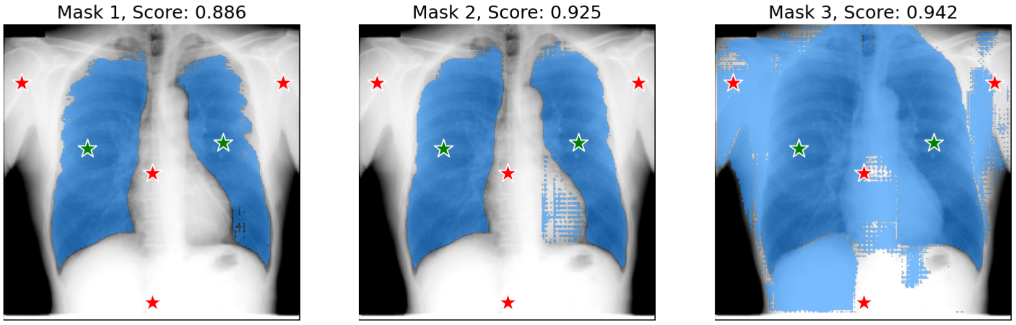
We engineer a prompting method using the *click prompt* method for prompting SAM. We calculate two green dots - which determine inclusion of mask(s) respective to the dot - and four red dots which determine exclusion.

We set the two green dots, by calculating the Center Of Mass (COM) of each ground truth mask in the training set, and then setting the two green *click prompt* coordinates to be the mean value of these COMs. This assumes a few things. First of all it assumes that the centering in each x-ray image is relatively uniform, which by browsing through the images in the data seems a reasonable assumption. Further it assumes that the mean COM of all lung masks in the dataset, should provide a true segmentation prompt.[1] There is no guarantee for this assumption. Lungs vary greatly in size, and lungs which deviate a lot from the mean in size, may get a false positive prompt.[2]

Four red dots a placed completely heuristically. These dots are set such that two are around the shoulders, one is in the center of the image, and one is centered on the x-axis, but low on the y-axis towards the groin. In the name of experimenting with *click prompting* we aren't aiming for statistically sound prompts, but rather to test the evaluation on simple and partly heuristic prompting. Similar unverified assumptions are made for setting these red dots. The results are shown in the figure below, for the first image in the training set.

[1] When the prompt correctly identifies and includes the lung masks (analogous to true positive).

[2] When the prompt incorrectly identifies and includes areas as lung masks where lungs are not present (analogous to false positive)



Predicted masks for engineered click prompts

Figure 3: Illustration of engineered click prompts on single image

In figure 3 we see the resulting predicted masks and scores for the predictor object, with `multi_mask=True` with the ambiguous prompting of our *engineered click prompts*. The documentation suggests setting multi mask to true, even when just one mask is needed. The claim is, that one can choose the highest scoring mask if just one mask is desired. Looking at the three masks, it seems[3] that masks one through three include increasing area of segmentation, which can be confirmed in the paper.[1][4].

The scoring is a result of learned internal scoring mechanism. It's essentially a predicted IoU. In the following section we will evaluate the usability of said scoring for our segmentation task and prompting method.

2.2 Evaluation of prompts

We've now engineered five *click prompts* for SAM. Two green (inclusive) dots and three red (exclusive) dots. These aren't dynamic prompts, but simply 5 sets of coordinates. We evaluate the prompts on the validation set, using the predictor object and reporting the F1 score for each predicted mask against ground truth mask. We keep multi mask set to true for evaluation, and for selecting which mask to evaluate on, we experiment with two approaches:

1. **Internal Scoring:** For the first approach we follow the recommendation of the documentation, and evaluate against the highest scoring mask of the three predicted masks for each image in the validation set.
2. **Heuristic Approach:** Following the observation that the third mask scores higher on the internal metric, yet outputs a useless mask(s), we will evaluate heuristically by always picking mask two from the three predicted masks.

We can than evaluate and compare the results of the two different approaches.

Internal Scoring Results

Internal Scoring; Mean F1 score: 0.7097

Internal Scoring; Standard Deviation on F1 score: 0.1523

Heuristic Approach Results

Heuristic Approach; Mean F1 score: 0.9009

Heuristic Approach; Standard Deviation on F1 score: 0.0256

[3] Judging by examination across multiple plots similar to figure

[4] See Appendix A page 17

We see a significant difference in the scoring. This supports our claim that the moderate segmentation amount amongst the predicted masks (mask 2) is a better fit to our task, than relying on the internal scoring. The internal scoring also makes underlying assumption, and tries to abstract away quality scoring to a learned scoring metric.

In conclusion, our simple prompting method requires very little implementation, and provides relatively decent results. SAM meets its claim about zero-shot prompting capabilities. However the scoring is not universally reliable, and considerations should be made before relying on these.

3 Dynamic Prompting

3.1 Bounding Boxes

For generating the bounding boxes from the ground truth label set from the provided dataset, we tackle it with a rather straightforward approach. From each ground truth mask, we extract the "coordinate" of the minimum x- and y-coordinate, as well as the maximum, where by coordinate we refer to an integer between 0 and 256 since the images are 256×256 pixels. By doing so, we can make a bounding box where these coordinates specify the top left corner and the bottom right corner. We make sure to respect the axis orientation in the code when implementing this calculation, and we perform the calculation twice. Once for the left side and once again for the right side, resulting in two bounding boxes per image (one for each lung).

3.2 Prompt SAM with Boxes

By producing the bounding boxes for the validation set using the function described above, we can feed the predictor object with these boxes to produce segmentation masks within the boxes. In doing so we get the following scores on the validation set:

> Bounding Boxes on Validation Set Mean F1 score: 0.9218
>
> Bounding Boxes on Validation Set Standard Deviation of F1 score: 0.0026

Which outperforms our *click prompting* method. It is important to note, that the ground truth masks are used for determining the bounding boxes, whereas the engineered prompting method only used the ground truth masks of the training set. In the next section we explore an approach to predict these bounding boxes given only the training and validation data, and evaluate the ability to generalize by scoring this dynamic prompting approach on the test set.

3.3 YOLOv8 Box Prediction and prompting SAM

Model Selection

There are several powerful pre-trained object detection models available in 2024. For this task of predicting bounding boxes, we will train YOLOv8. YOLOv8 is the eigth edition of an object detection model by ultralytics, where YOLO is an acronym for You Only Look Once. This model is designed for fast segmentation, and real-time performance. YOLO models are known for their simple yet effective architecture. While the original YOLO had a straightforward structute with a single convolutional layer, followed by a fully connected layer, YOLOv8 introduces some improvements.

A feature of YOLOv8 is the anchor-free detection mechanism. This allows for predicting the center of an object instead of relying on predefined anchor boxes. This makes YOLOv8 one of the lightest models capable of benchmark performance, maintaining high accuracy while offering real-time inference speeds on modern GPUs.[2]

Why choose YOLOv8? In the spirit of testing the potential of the segmentation foundation model SAM with its real-time segmentation capabilities, making fast predicted bounding boxes for prompting, poses an interesting angle. We will test and see, if light training on a light model[5] can introduce satisfactory

[5] Personal training time for 10 epochs was less than two minutes on a 4 year old laptops GPU

results.

Preprocessing and Training

In order to be able to train the model, we have to preprocess the data a bit. For the model to learn, we must provide the ground truth bounding boxes in the format:

$$(< classid >, < x\_center >, < y\_center >, < width >, < height >).$$

We assign a class for left and right lung, and write a function which converts the ground truth bounding boxes to the required format. After saving the data to a .yaml file, we can train the model.

The model is trained just once, on 10 epochs. No sweeping or hyperparameter selection is conducted, since this is a means to test a method for dynamic prompting. After 10 epochs training shows the following results:

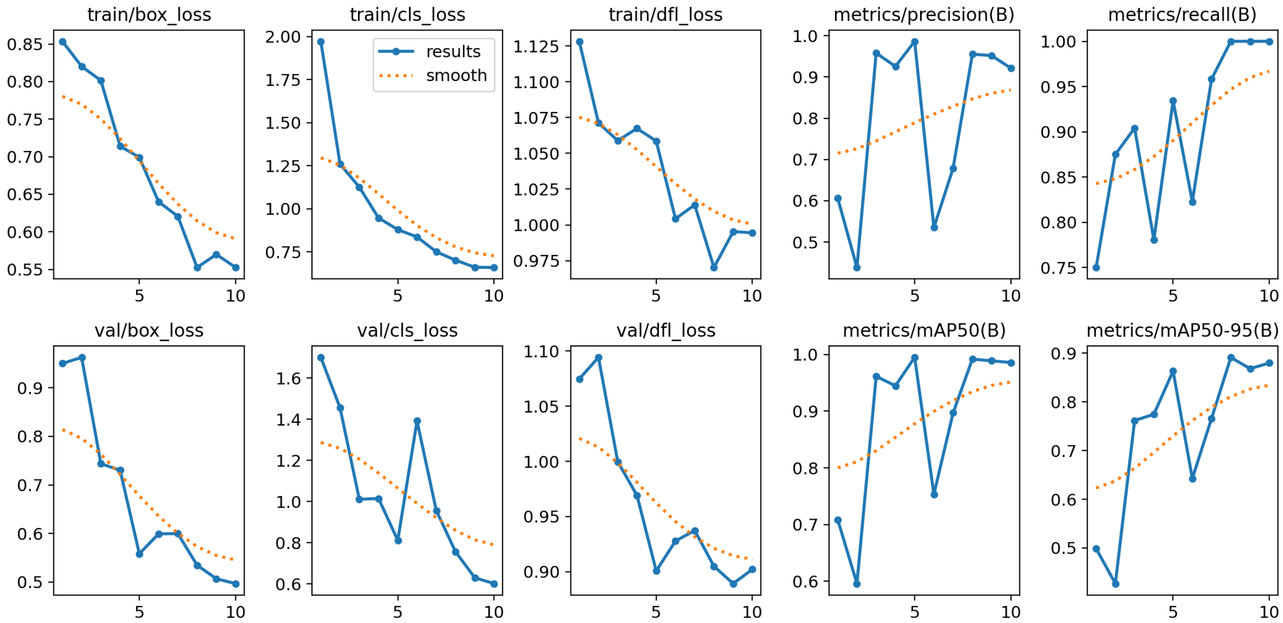


Figure 4: Different loss metrics, precision, recall and mAP50 results from training YOLOv8

On just 10 epochs, we see definite training, and most metrics (including various loss, precision, and recall) seem to be approaching convergence based on visual inspection.

The results indicate, that the model is effectively learning and should be able to provide reliable bounding boxes for the subsequent promping task. The training performance indicates the models ability to generalise unto unseen data, and suggests that it will suffice to provide bounding boxes for prompting SAM on unseen data.

We show three randomly sampled images, with their ground truth bounding boxes and the predicted ones by the trained version of YOLOv8, to evaluate the results:

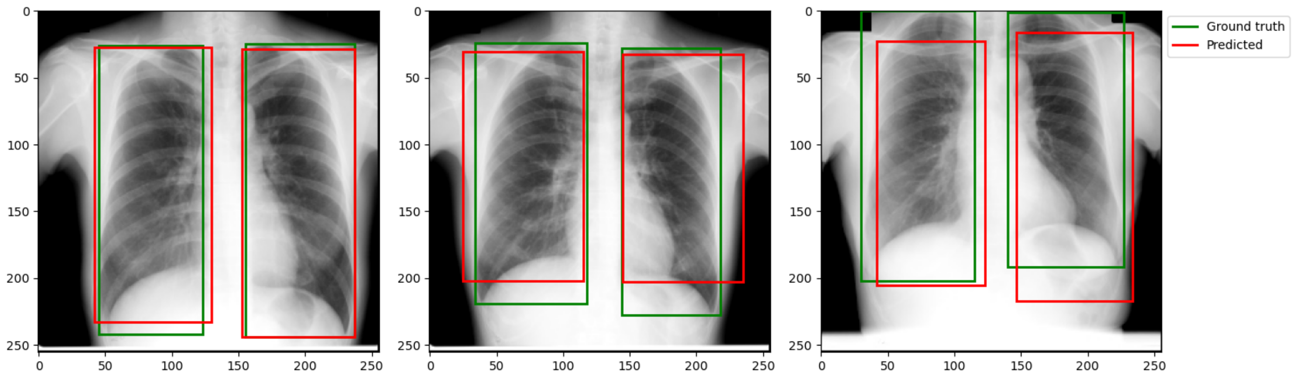


Figure 5: Randomly sampled images for comparing predicted vs GT Bboxes

The samples supports the belief that the model has trained.

Evaluating Predicted Bounding Boxes

We now present the mean IoU score of predicted bounding boxes against ground truth bounding boxes in the test set:

Mean IoU score on test set: 0.7044

Which is somewhat lower than desired, but still shows that the trained version of YOLOv8 shows some capability to generalize unto unseen data.

For further examining the generalization, we plot the image with highest and lowest IoU score:

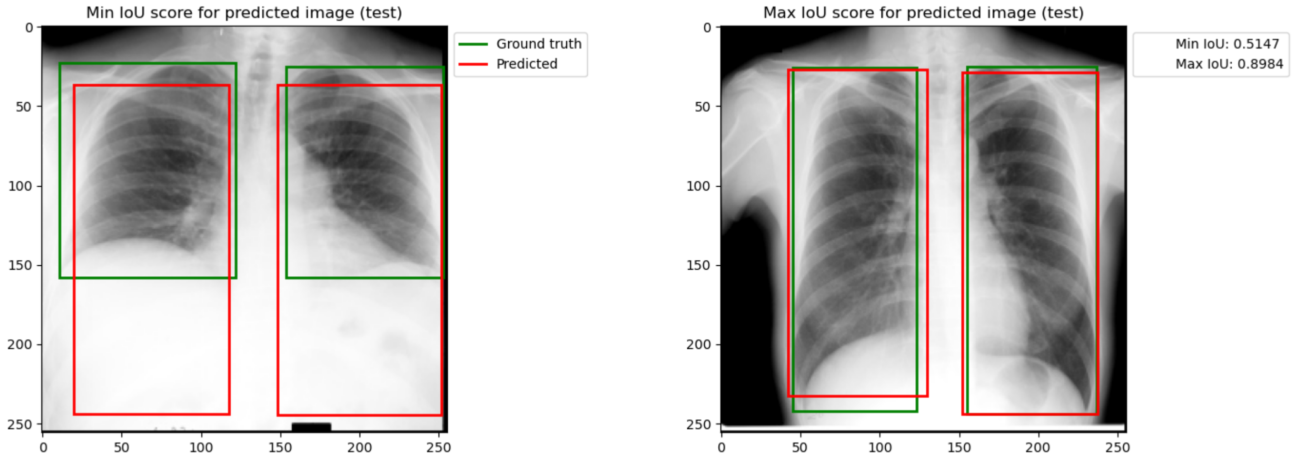

Min IoU score for predicted image (test)
Max IoU score for predicted image (test)
Ground truth
Predicted
Min IoU: 0.5147
Max IoU: 0.8984
0
50
100
150
200
250

Figure 6: GT vs predicted Bboxes on best and worst IoU scoring test set predictions

The worst performing prediction seems to exhibit a test subject with rather small lungs, whereas the best performing exhibit the opposite. Having experimented with SAM, I believe the segmentation for the random samples and the best predicted bounding box will be descent, whereas the mask resulting from the worst performing predicted bounding box is dubious. But we shall see.

We now prompt SAM with the predicted bounding boxes provided by YOLOv8 on the test images, and evaluate the resulting masks predicted by SAM. We are setting multi masks to false.

The ground truth bounding boxes yield a mean F1 score of 0.9218 with a standard deviation of 0.0026. Since our predicted bounding boxes are trained on these ground truths, we expect the resulting mean F1 score to be at most this value. Achieving a similar mean F1 score would indicate that our predicted boxes perform comparably well to the ground truth boxes on the validation set. Exceeding this score would suggest that our predicted boxes perform better on unseen data than on training data, which is highly unlikely.

We prompt sam with the predicted bounding boxes on the test set[6] and report mean F1 score and standard deviation:

Dynamic Prompting F1 Score

Dynamic Prompting Mean F1 Score: 0.8828

Dynamic Prompting Standard Deviation of F1 Score: 0.0735

Evaluating and Comparing Final Results

The resulting mean F1 score has a relative difference of about [7] 4.23% compared to the ground truth box prompting, which is quite satisfactory. The standard deviation differs more on the other hand, where the Relative Standard Deviation (RSD) of the dynamic prompting is about[8] 8.33% against 0.28%, which is a drastic difference. It is also relevant to consider the sample sizes, since a tenfold test size increases the likelihood of outliers, thus the solid RSD of 0.28% is a less reliable measure due to the small validation set size.

Despite higher variability, the overall results show that our dynamic prompting is a success. The method shows good generalization, on the large test set, effectively using the predicted bounding boxes to prompt SAM for segmentation.

Comparing with U-net

Comparing the results from assignment 1, for our previous U-net sweep, the best performing model had a mean F1 score of 0.7263 on the test set, with a standard deviation of 0.0847. Comparing these results with our dynamic prompting approach, we had a mean F1 score of 0.8828 with a standar deviation of 0.0735.

The significant improvement in F1 scores suggests that the dynamic prompting method with SAM outperforms U-net. One possible explanation could be that SAM has better generalisation potential unto unseen data, due to its pre-training on a large and diverse dataset. This result is particularly interesting, since U-net is known to be excellent for medical image segmentation tasks.

Reliability for Medical Image Segmentation

This method may not be fully reliable for medical image segmentation, due to the high precision required for such. However, what's striking about this experiment, is how little training and implementation is needed, to get this degree of quality segmentation masks. Further improvements could definitely be made, such as better training, additional/alternate object detection methods or combining different prompting techniques.

Final conclusions

In conclusion, dynamic prompting with SAM demonstrates impressive results and provides a lightweight, easy-to-use option for segmentation tasks. While further overall improvements and training is needed to match the precision demanded for medical application, the current performance indicates strong potential. Exploring different object detection models, combining prompting methods and further training could enhance the efficacy of this approach.

[6] Note the test set is of size 123 whereas the validation set is of size 12

[7] $\frac{0.9218-0.8828}{0.9218} \approx 0.0423$

[8] $RSD = \frac{STD}{Mean}$

References

[1] Alexander Kirillov, Eric Mintun Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, Ross Girshick. *Segment Anything*, 2023.

[2] Dillon Reis, Jacqueline Hong, Jordan Kupec, Ahmad Daoudi. *Real-Time Flying Object Detection with YOLOv8*, 2024.