



Latest updates: <https://dl.acm.org/doi/10.1145/3674658.3674664>

RESEARCH-ARTICLE

Adaptive Segment Anything Model for Spatial Transcriptomic Cell Segmentation

YUXIN PU, University of Electronic Science and Technology of China, Chengdu, Sichuan, China

Open Access Support provided by:

[University of Electronic Science and Technology of China](#)



PDF Download
3674658.3674664.pdf
05 January 2026
Total Citations: 0
Total Downloads: 152

Published: 24 May 2024

[Citation in BibTeX format](#)

ICBBT 2024: 2024 16th International Conference on Bioinformatics and Biomedical Technology

May 24 - 26, 2024
Chongqing, China

Adaptive Segment Anything Model for Spatial Transcriptomic Cell Segmentation

Yuxin Pu*

202221080301@std.uestc.edu.cn

School of Computer Science and Engineering, University of Electronic Science and Technology of China
Chengdu, Sichuan, China

ABSTRACT

With the development of spatial transcriptomics, now we could have a much better understanding of the spatial genomic profile of complex tissue. However, the cell segmentation remains a critical step for data analysis processing. Here we developed an adaptive SAM model for cell segmentation in the spatial transcriptomics. SAM model has been developed as a powerful basic model for segmentation. In the application of biomedical image segmentation, SAM did not perform equally well for different dimension of images. In our study, we first exhibited that SAM model segmentation is sensitive to image dimension. We provided minimal human annotation to initiate the optimization for finding the best dimension images for SAM model. We found that adaptiveSAM could perform better or equally well in the biomedical image segmentation without finetuning, compared with Faster RCNN and Mask RCNN based detection head.

CCS CONCEPTS

- Computing methodologies → Supervised learning.

KEYWORDS

Cell Detection, Cell Segmentation, Segment Anything, Faster RCNN, Mask RCNN

ACM Reference Format:

Yuxin Pu. 2024. Adaptive Segment Anything Model for Spatial Transcriptomic Cell Segmentation. In *2024 16th International Conference on Bioinformatics and Biomedical Technology (ICBBT 2024), May 24–26, 2024, Chongqing, China*. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3674658.3674664>

1 INTRODUCTION

Spatial transcriptomics has now been a fast-developing field [6]. One critical step in the spatial transcriptomics is the cell bin segmentation [8]. For sequencing-based approaches, nuclei image segmentation could provide critical information for cell bin segmentation. Accurate segmentation of nuclei has become a bottle neck for a lot of application in image-based approaches. At the same time, abundant effective algorithms of machine learning in computer

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICBBT 2024, May 24–26, 2024, Chongqing, China

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1766-6/24/05
<https://doi.org/10.1145/3674658.3674664>

vision field have been proposed one after another to achieve more efficient and accurate object detection results. With all these techniques, it is inevitable to raise the idea of auto cell segmentation method, as it is tedious to analyze relevant biological features such as cell count, type, division, shape, etc. manually. However, with the help of auto cell segmentation methods based on computer vision technology, scientists are able to evaluate how relevant biological features change over time and in response to a variety of conditions efficiently and effortlessly. Therefore, methods for reliable cell segmentation are of great research value.

Developed from Fast RCNN [2], Faster RCNN [9] is proposed to achieve real-time object detection with Region Proposal Networks (RPN) which replaces the Selective Search in RCNN and Fast RCNN to generate proposal regions. Faster-RCNN consists of four modules, which are the backbone, RPN, RoI and Fully Connected Networks. Specifically, the input image is first fed to backbones to extract features and then obtain the shared feature map. Second, feature map as the input of the RPN module goes through RPN convolution network, then positive anchors and corresponding bounding box regression offsets are calculated, next the proposals are generated. Then, with the shared feature map, proposals are further selected in RoI Pooling layer. Finally, the proposal features are sent to subsequent fully connected networks for box regression and classification.

Segment Anything Model (SAM) [5] consists of three components, which are the image encoder computing an image embedding, the flexible prompt encoder embedding prompts, and the fast mask decoder predicting segmentation masks. For the image encoder, SAM uses an MAE [3] pre-trained Vision Transformer (ViT) [1] minimally adapted to process high dimension inputs. For prompt encoder, SAM represents points and boxes by positional encodings summed with learned embeddings for each prompt type and represents free-form text with an off-the-shelf text encoder from CLIP [7]. Dense prompts (i.e., masks) are embedded using convolutions and summed element-wise with the image embedding. For mask decoder, its design employs a modification of a Transformer decoder block [10] which uses prompt self-attention and cross-attention in two directions, and a dynamic mask prediction head is followed to computes the mask foreground probability at each image location.

In this study, we aim to develop an adaptive SAM model to enable it to detect nucleus for spatial transcriptomics as well as wider application in tissue segmentation and compare it with Faster RCNN based models. In order to train the model, a dataset is required. Since the raw images of high-dimension fluorescence microscopy have no labels and cannot be used for direct model training, the raw data need to be preprocessed. Our data preprocessing has four stages: image cropping, data annotation, data augmentation and

data format normalization. In the first stage, we crop the large sized raw images to smaller tiles and select tiles with normal cell distribution as the training data. In the second stage, we first utilize Labelme annotation tool to generate bounding boxes for cells on selected tiles and then we input the bounding box annotation to SAM model to generate masks for those cells. In third stage, applying a combination strategy, the annotated data is used to generate synthetic images for data augmentation purpose. In final stage, all data we obtain is formalized to COCO data format which model can process. After data preprocessing, we use our preprocessed high dimension cell image dataset to fine-tune the Faster RCNN and Mask RCNN [4]. With trained Mask RCNN, we can directly segment cells on input images. With trained Faster RCNN, since we can only directly obtain the bounding boxes of cells on input images, we further exploit SAM model with prompting the obtained bounding boxes to segment cells. To decide a better method for cell segmentation in this study, we also make the comparison on cell segmentation results in trained trained Faster RCNN+SAM and SAM.

2 RELATED WORKS

2.1 Image Cropping for Efficient Handling

High-dimensional images were processed to make them more manageable for subsequent tasks. These images were cropped into smaller tiles, each measuring 41x41 pixels. The dimension was chosen to ensure that significant features within the images were retained without compromising computational efficiency.

2.2 Manual Annotation using Labelme

For precise object detection and segmentation, bounding boxes around individual cells were manually drawn. We utilized the Labelme annotation tool, a graphical image annotation tool written in Python, for this purpose. Labelme allows for meticulous labeling, ensuring that each cell is accurately boxed.

2.3 Bounding Box Information Integration

Once the bounding boxes were established, their coordinates and related information were fed into the "Segment Anything" model. This step ensures that the model receives accurate data points for cell segmentation.

2.4 Segmentation using RLE Encoding

The "Segment Anything" model was employed to segment the images. The output from this model was in the form of Run-Length Encoding (RLE). RLE is a compact way to represent the segmented areas, where continuous groups of pixels are encoded as a single data point, thereby compressing the representation.

2.5 Conversion to COCO Format

Subsequently, the RLE-encoded segmentation data was transformed into the Common Objects in Context (COCO) format. This is a widely accepted format in the computer vision community and facilitates interoperability with various modeling tools and frameworks.

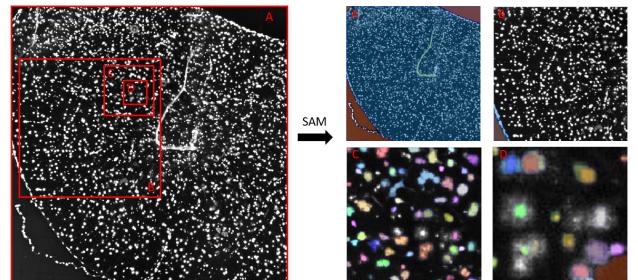


Figure 1: Generic SAM model is sensitive to input dimensions. Left is the original image with dimension 1000×1000 pixels, the right are four different images with equal or smaller dimension: (A) Image with dimension of 1000×1000 pixels and the segmentation results from SAM; (B) Image with dimension of 500×500 pixels and the segmentation results from SAM; (C) Image with dimension of 160×160 pixels and the segmentation results from SAM; (D) Image with dimension of 60×60 pixels and the segmentation results from SAM.

2.6 Visualization of Segmentations

To ensure the quality of the segmentation, we visualized the segmented images. This step involved overlaying the segmentation masks onto the original images to check for alignment and accuracy. Any inconsistencies or errors detected at this stage were addressed before moving on.

2.7 Data Splitting for Model Training

The data set was divided into training and validation subsets to train and evaluate the model's performance. In our preliminary attempt, 72 images were designated for training, while 9 were set aside for validation.

2.8 Model Training using Mask RCNN

With the data prepared, we proceeded to train our model using Mask RCNN, a prominent model for object detection and instance segmentation. This model is known for its capability to effectively distinguish objects in images and generate precise masks around them. Our training regimen was adjusted based on initial results and iterative feedback from the validation set.

3 METHOD

As Segment Anything Model is a powerful foundation model which can be generalized on various tasks. Inspired by this, we come up with an idea that if it is feasible to introduce the SAM model directly to handle with the cell segmentation task. If the idea is verified, instead of fine-tuning traditional image segmentation model with extra task-specific modeling expertise, the cell segmentation can be accomplished easily by utilizing the SAM model. To this end, we conducted an experiment in which we tested the SAM model on cell images with different dimension. Fig. 1 illustrates the results on cell images with four different dimensions and all segmentation results were generated by using the segment everything mode of SAM. It is obvious that with the increase of input image dimension,

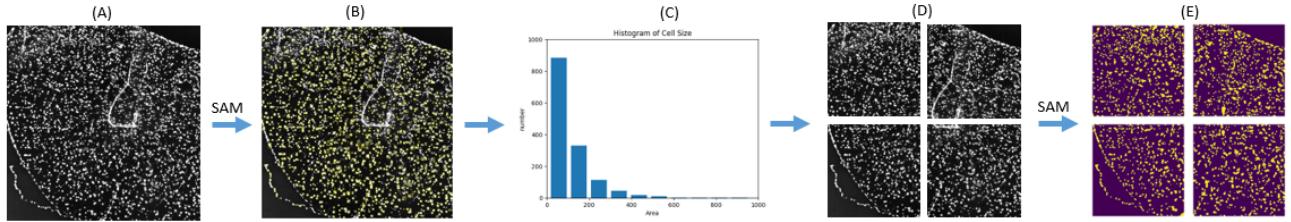


Figure 2: The process of SAM-in-SAM: (A) The original image whose size is 1000×1000 ; (B) The output from SAM on which yellow rectangle represents the detected cell on the original image; (C) The histogram's x-axis is the cell size and y-axis is the number of cells; (D) The cropped images whose size are 500×500 ; (E) The output from SAM which yellow mask represents the segmentation of each cell on cropped images.

the model would focus more on the overall structure rather than fine grade details, such as nuclei segmentation.

3.1 SAM-in-SAM

As the generic SAM model is not working perfectly in nuclei segmentation and sensitive to input dimension, we hypothesize that we could use SAM-in-SAM, which refers to the coarse-to-fine process by applying SAM model twice. As illustrated in Fig. 2, we first apply SAM model for initial segmentation, with a much lower accuracy but able to predict optimized parameter (cell size). Then we apply SAM again on cropped inputs, which is obtained with the help of optimized parameter, to obtain more accurate results.

Specifically, by feeding the original image Fig. 2 (A) to SAM model, though with a much lower accuracy, an image of detected cells Fig. 2 (B) is obtained. Then, with the information of output from SAM, the histogram of detected cells' size Fig. 2 (C) can be generated. From the histogram, we can predict a most representative cell size for the original image. After this, we multiply the cell size with a scale factor s to get the total number of pixels on cropped image with the most proper size denoted as d , so the dimension of cropped image d can be calculated by the following formula:

$$d = \sqrt{\text{cell_size} \times s} \times \sqrt{\text{cell_size} \times s} \quad (1)$$

Then, we crop the original image to smaller tiles with dimension of d . Finally, the cropped images are fed to SAM again to get the masked images on which all cells are segmented precisely.

As illustrated in Fig. 3, we conduct many experiments to decide the scale factor s . Specifically, the original image 3 (A) is first input to SAM model. Then, based on the output of SAM, the histogram of cell size 3 (B) is generated, from which most representative cell size is 50. To decide a proper value for scale factor s , we test a lot of values. Here, we only demonstrate the final segmentations 3 (C), 3 (D) and 3 (E) with value of 200, 5000 and 15000. It is obvious that 5000 is the largest value of s to maintain the accuracy of segmentation from SAM. Additionally, as illustrated in Fig 4, we further verify the idea of using this model for other medical images like tissue and brain image.

As shown in Fig. 1, it is impossible to obtain accurate location and segmentation of each cells by simply applying the SAM model once. However, as shown in Fig. 3, with the fixed parameter s we found, after applying the SAM model twice, we can obtain

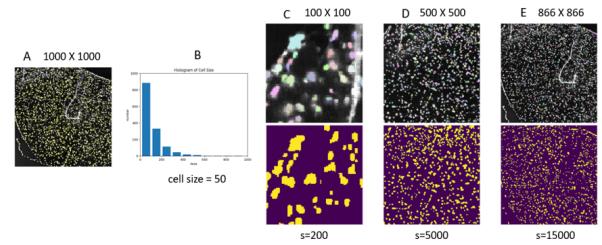


Figure 3: The process of deciding the value of scale factor s : (A) The original image with dimension of 1000×1000 ; (B) The histogram of detected cells' size from which the most representative cell size is 50; (C) The outputs from SAM on the cropped image with dimension of 100×100 ; (D) The outputs from SAM on the cropped image with dimension of 500×500 ; (E) The output from SAM on the cropped image with dimension of 866×866 .

accurate location and segmentation of almost all cells. Therefore, proposed method can improve the accuracy of SAM model in cell segmentation in complex tissues, and the key of this method is to use the fixed parameter s and apply the SAM model twice.

3.2 Further Analysis for SAM-in-SAM

To further analyze the performance of SAM-in-SAM, we also compare the accuracy of SAM-in-SAM model with Faster RCNN+SAM and Mask RCNN. In order to adapt Faster RCNN and Mask RCNN to the cell detection and segmentation task, the pre-trained models need to be fine-tuned by using cell dataset. Since the raw images of high-dimension fluorescence microscopy have no labels and cannot be used for direct model training, the raw data need to be preprocessed to generate cell dataset. Our data preprocessing has four stages: image cropping, data annotation, data augmentation and data format normalization. In the first stage, we crop the large sized raw images to smaller tiles and select tiles with normal cell distribution as the training data. In the second stage, we first utilize Labelme annotation tool to generate bounding boxes for cells on selected tiles and then we input the bounding box annotation to

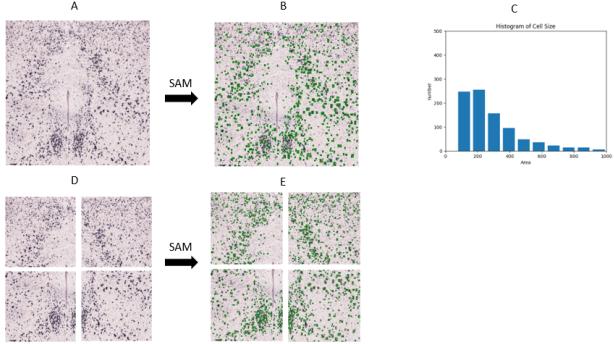


Figure 4: The segmentation results from SAM-in-SAM by using brain images: (A) The original image with dimension of 2000×2000 ; (B) The output from SAM on which green rectangle represents the detected cell on the original image; (C) The histogram of detected cells' size whose x-axis is the cell size and y-axis is the number of cells; (D) The cropped images with dimension of 1000×1000 .

SAM model to generate masks for those cells. In third stage, applying a combination strategy, the annotated data is used to generate synthetic images for data augmentation purpose. In final stage, all data we obtain is formalized to COCO data format which model can process.

3.2.1 Training Data Annotation. In this work, we also trained a customized Faster RCNN model and Mask RCNN model specifically for nuclei segmentation. To achieve this goal, we first manually annotate the data. By utilizing Labelme annotation tool, the rectangle bbox of each cell can be drawn on the original image Fig. 5 (A) to get the labelled image Fig. 5 (B). Then we got json files as the output of Labelme to save the annotation. Secondly, we feed the json file of the annotation into the Segment Anything Model (SAM) to get the mask of each cell in the image Fig. 5 (C), and we also superimpose the mask of all cells to get a superimposed image Fig. 5 (D).

Initially, we only annotated one image as the dataset. However, with the experiment carrying on, the result is not satisfying. As illustrated in Fig. 6, we adopt the workflow above and annotate 5 more images by utilizing the Labelme annotation tool, and we

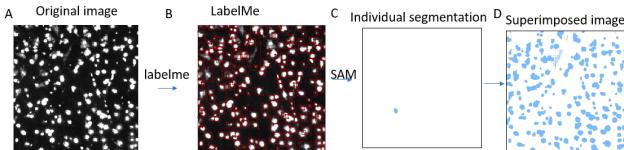


Figure 5: Manual annotation of nuclei image: (A) The original image; (B) The red rectangle represents the manually labelled bbox; (C) The blue mask represents the segmentation of the individual cell in the image; (D) The superimposed blue masks represent the segmentation of all the cells in the image.

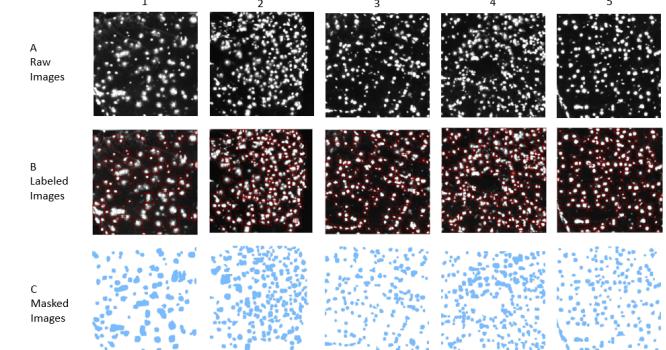


Figure 6: Labeled images and masked images of five raw nuclei images: (A) Five raw nuclei images; (B) Five labeled images by utilizing the Labelme annotation tool; (C) Five masked images from Segment Anything model.

input the json file storing annotation of each raw image to SAM to get the mask image for each raw image.

As there are many cells overlap with each other, it is difficult for human eyes to identify the edge of these cells. To tackle with this problem, we applied two filters to preprocess the images before annotating the images. Specifically, the Gaussian filter is first applied to blur the raw image, and then the Laplacian filter is applied later to enhance the edge information, so that most of the edge of overlapping cells can be identified when annotating the edge-enhanced image. As illustrated in Fig. 7, we can easily find that more accurate bounding box are drawn on the edge-enhanced image than on the raw image.

3.2.2 Data Augmentation with Synthetic Images. As illustrated in Fig. 8, we tried to use only one annotated images to train the model, thus to increase the dataset without annotating more image, the superimposed masked image from SAM and the original image are

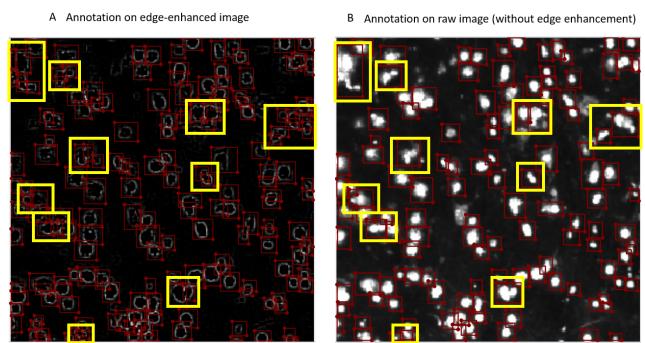


Figure 7: Annotation comparison between edge-enhanced image and raw image: (A) Annotation on edge-enhanced image on which yellow rectangles are more accurate bounding boxes of the cells in the same areas on the raw image; (B) Annotation on raw image on which yellow rectangles are less accurate bounding boxes of the cells in the same areas on the edge-enhanced image.

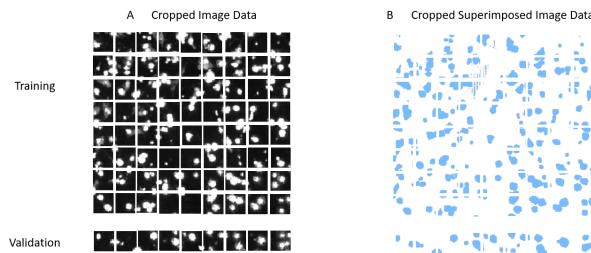


Figure 8: Cropped images from one image: (A) The original image cropped into 81 smaller tiles which are divided as 8:1 of training and validation dataset; (B) The superimposed masked image cropped into 81 smaller tiles which are divided as 8:1 of training and validation dataset.

cropped into 81 smaller tiles (41×41 pixels) respectively which are divided into 72 images of training dataset and 9 images of validation dataset. However, the dimension of 41×41 pixels is much lower dimension which easily leads to low accuracy of detection when utilizing the trained model in some higher dimension images, which was proved in our experiment.

As models trained with small-sized tiles of image has worse capability when being applied to some larger dimension images, we come up with the idea of using synthetic images to process data augmentation. In other words, the already cropped images can be combined in some simple procedures to generate more larger dimension images. Before synthesizing images, few procedures are involved. Specifically, to generate a large number of different combinations of synthetic images, more annotated images are needed which ensure we can obtain enough higher dimension cropped images as the synthesizing materials. This explains the reason why we annotated another 5 images. Then, we cropped each pair of raw image and masked image to get smaller tiles (123×123 pixels). With these smaller tiles, we can combine every two of them to synthesize images with larger dimension. As shown in 9, take Image 1 from Fig. 6 as an example. We first cropped the raw image and masked

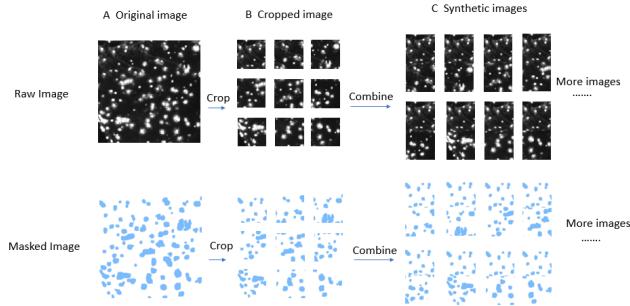


Figure 9: The process of synthesizing higher dimension images: (A) The original raw image (or masked image); (B) The cropped images from raw image (or masked image). (C) The synthetic images of raw images (or masked images).

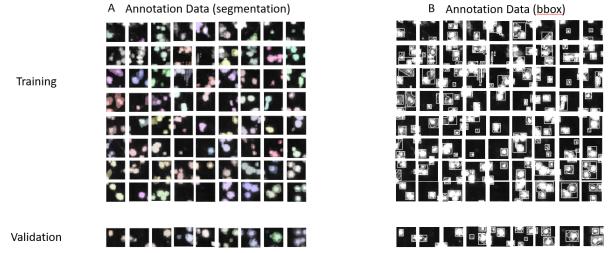


Figure 10: Visualized segmentation and detection from SAM: (A) The colorized region represents the segmentation of each cell in cropped images from SAM; (B) The white rectangle bbox represents the detection of each cell in cropped images from SAM.

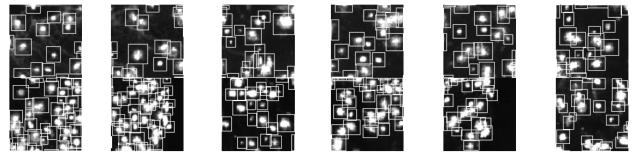


Figure 11: The visualization from generated COCO format json file of synthetic images.

image into 9 smaller tiles, and then we use a simple combination strategy to combine every two smaller tiles from raw images vertically. Additionally, the same combination strategy is also applied to the smaller tiles from masked images. In this way, we get the higher dimension images and their corresponding masked images. Additionally, we also divide the synthetic images to training dataset (924 images), validation dataset (396 images) and test dataset (396 images). Additionally, to make the model perform well on the image dimension which is larger than 123×246 pixels, we also directly combine the original images to generate synthesized images whose dimension is 369×738 pixels. In the experiment with the edge enhancement preprocessing, the edge-enhanced images are under the same process following the same flow-work mentioned above.

The masked images obtained from above procedures need to be processed to store cells' corresponding information in the COCO format json file which is a kind of dataset needed to feed the later model. To obtain the COCO dataset, each cropped mask image is binarized, so the edge information and the location information of the cells can be extracted, and with the paired cropped original image the needed information can be saved to json file in COCO format. Importantly, as illustrated in Fig. 10 and Fig. 11, to ensure the correctness of the detection and segmentation information from SAM, we visualize the COCO format json file to show the segmentation and the bbox of each cell (the edge-enhanced images are under the same process following the same flow-work).

4 EXPERIMENT

We adopt mmdetection to train and test models on CPU with a batch size of 2. The start learning rate is set to 0.001. Other hyper-parameters are the same to the default settings of each detector. The Faster RCNN was fine-tuned with 702 training images and 72 validation images. The Fig. 12 shows the change curves of the accuracy (Fig. 12 A) and the losses (Fig. 12 B) during model training.

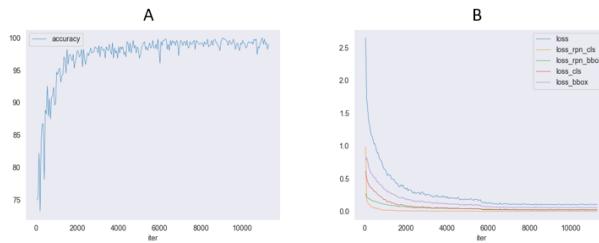


Figure 12: The change curve of accuracy A and losses B in training Faster RCNN.

After fine-tuning the Faster RCNN on cell dataset, we tested the trained model by using 72 images. As shown in Fig. 13, it is obvious that Faster RCNN achieve a quite accurate detection on cell image we provided. What's more important is that even with the large dimension cell images, fine-tuned Faster RCNN can achieve accurate prediction on the cell images, as long as the dimension of the image is the same as the training images. Additionally, with accurate detection from Faster RCNN, SAM can have precise segmentation on cell images without doubts. Finally, we can compare these models and give a conclusion that SAM-in-SAM is doing equally to even better compared with Faster RCNN+SAM in nuclei segmentation. As illustrated in Fig. 14, it's obvious that even without any quantitative comparison metrics or statistical analyses when using the SAM-in-SAM model and Faster RCNN + SAM model respectively on the same input image, almost every cell on the input image are detected when using the SAM-in-SAM model, whereas less than half of the cells are detected when using the Faster RCNN + SAM model.

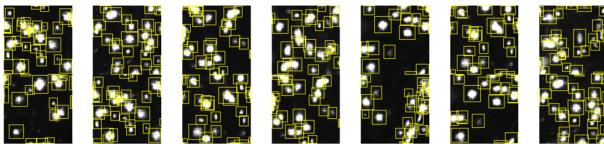


Figure 13: The predictions from trained Faster RCNN model.

4.1 Discussion

In this study, we aim to develop a most effective method to detect nucleus in spatial transcriptomics accurately. As Segment Anything Model is a powerful foundation model which can be generalized on many tasks, we conducted the experiment to figure out whether SAM model can achieve our goal directly. During the experiment,

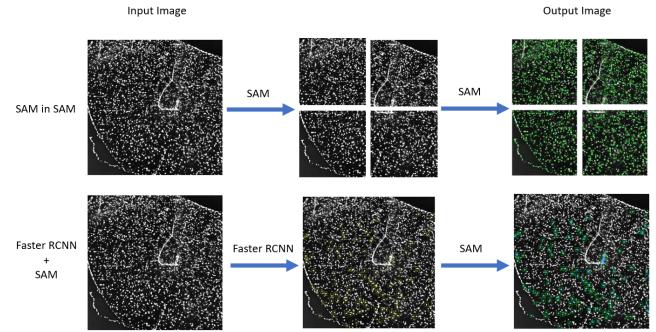


Figure 14: Performance comparison of SAM-in-SAM and Faster RCNN+SAM.

we find that SAM is sensitive to the dimension of the image, which means with the increase of input image dimension, the model would focus more on the overall structure rather than fine grade details. To deal with this problem, we propose the idea of SAM-in-SAM which uses SAM model twice to obtain the accurate segmentation from the image. Furthermore, to verify the idea, we compare SAM-in-SAM with Faster RCNN based models. We found that our trained Faster RCNN has a low detection accuracy on the test image whose size is 1000×1000 and the reason for this is that we fine-tuned Faster RCNN with images whose dimension is smaller than the test image. This problem can be fixed in the future study by enlarging dataset with images of more various dimensions. This is to say, to have a better segmentation result with Faster RCNN based model, we need to generate more training data. Therefore, it can be concluded that SAM-in-SAM is doing equally to even better compared with Faster RCNN-based model fine-tuned with small dataset in nuclei segmentation. The biggest advantage of SAM-in-SAM over Faster RCNN-based model is that there is no training process. By calculating the proper size of cropped images with the fixed scale factor s , SAM can achieve high segmentation accuracy on these cropped images.

REFERENCES

- [1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).
- [2] Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 1440–1448.
- [3] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 16000–16009.
- [4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.
- [5] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4015–4026.
- [6] Alona Levy-Jurgenson, Xavier Tekpli, Vessela N Kristensen, and Zohar Yakhini. 2020. Spatial transcriptomics inferred from pathology whole-slide images links tumor heterogeneity to survival in breast and lung cancer. *Scientific reports* 10, 1 (2020), 18802.
- [7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

- [8] Anjali Rao, Dalia Barkley, Gustavo S França, and Itai Yanai. 2021. Exploring tissue architecture using spatial transcriptomics. *Nature* 596, 7871 (2021), 211–220.
- [9] Shaoging Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).
- [10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).