A systematic evaluation of computation methods for cell segmentation

Yuxing Wang[1,2,*], Junhan Zhao[3,4,*], Hongye Xu[1,*], Cheng Han[1], Zhiqiang Tao[1], Dongfang Zhao[5], Dawei Zhou[6], Gang Tong[7], Dongfang Liu[1,†], and Zhicheng Ji[2,†]

[1]Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY, USA.

[2]Department of Biostatistics and Bioinformatics, Duke University School of Medicine, Durham, NC, USA.

[3]Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA.

[4]Department of Biostatistics, Harvard T.H.Chan School of Public Health, Boston, MA, USA.

[5]Department of Computer Science & eScience Institute, University of Washington, Seattle, WA, USA.

[6]Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA.

[7]Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA.

[*]These authors contributed equally

[†]Corresponding author. E-mail: dxleec@rit.edu and zhicheng.ji@duke.edu

ABSTRACT

Cell segmentation is a fundamental task in analyzing biomedical images. Many computational methods have been developed for cell segmentation, but their performances are not well understood in various scenarios. We systematically evaluated the performance of 18 segmentation methods to perform cell nuclei and whole cell segmentation using light microscopy and fluorescence staining images. We found that general-purpose methods incorporating the attention mechanism exhibit the best overall performance. We identified various factors influencing segmentation performances, including training data and cell morphology, and evaluated the generalizability of methods across image modalities. We also provide guidelines for choosing the optimal segmentation methods in various real application scenarios. We developed Seggal, an online resource for downloading segmentation models already pre-trained with various tissue and cell types, which substantially reduces the time and effort for training cell segmentation models.

INTRODUCTION

Cell morphology, which describes the shape, size, and structure of cells, is crucial for understanding the types and states of cells and how cells respond to their microenvironments[1–7]. The study of cell morphology is enabled by biomedical imaging technologies such as phase-contrast microscopy[8] and multiplexed fluorescence imaging[9]. However, these technologies do not directly provide the boundaries of cells, and the cell boundaries must be defined either manually or by computational methods. While manually identifying cell boundaries is infeasible for a large number of images, computational methods for cell segmentation become indispensable for delineating cell boundaries in most settings. Many downstream analysis tasks, such as modeling sub-cellular patterns[10,11], screening genetic or chemical perturbations[12,13], tracking the cell migration and

development[14,15], and deciphering cell physiological status[16,17], depend on the computationally identified cell boundaries. Thus, an accurate computational method for cell segmentation is of paramount importance for appropriately handling biomedical images and extracting insights from the data.

The advancement of deep learning in the past decade has inspired numerous cell and image segmentation methods that significantly outperform traditional approaches, such as the watershed method[18]. These methods can be categorized into two classes. The first class includes methods developed and tested specifically for biomedical images, such as Cellpose[19], StarDist[20], RetinaMask[21], Mesmer[22], and FeatureNet[23]. The second class comprises methods like Centermask2[24], Mask RCNN[25], ResNeSt[26], MS RCNN[27], Mask2former[28], Cascade Mask RCNN seesaw[29], Swin Transformer[30], SOLOv2[31], RF-Next[32], HRNet[33], Res2Net[34], and Segment Anything[35], developed for more general purposes of instance segmentation. In principle, these general-purpose methods are applicable to biomedical images for cell segmentation, as it is a form of instance segmentation. However, they were primarily tested on generic datasets such as ImageNet[36], and their effectiveness in cell segmentation remains largely unexplored.

The performances of cell segmentation methods were compared in previous studies, including the Cell Tracking Challenge[37], the 2018 Data Science Bowl challenge[38], and a recent grand challenge organized by NeurIPS[39]. However, these studies have several limitations. First, the volumn of data included in these studies is limited. For instance, the 2018 Data Science Bowl provides 37,333 annotated nuclei, and the NeurIPS challenge provides 168,491 annotated cells. These numbers are substantially smaller than those of recently published datasets like LIVECell[8,40] and TissueNet[9,41], each containing more than a million annotated cells or nuclei. Since these large datasets are publicly available for training cell segmentation methods, the results of previous challenges no longer reflect the state-of-the-art performance of cell segmentation methods. Second, the performance of cell segmentation methods can vary dramatically across tissue types and imaging technologies. Previous benchmark studies focus only on the overall performance, averaged across various scenarios, and lack detailed evaluations for each tissue type and imaging technology. Since most segmentation tasks involve one tissue type using one imaging technology, the value of previous benchmark studies is limited in real-world practice for selecting the most appropriate segmentation method. Third, studies like the NeurIPS and Data Science Bowl challenges focus on evaluating methods submitted by challenge participants. These methods often lack open-source software packages with well-documented vignettes, making them difficult to apply to new datasets, especially for users without extensive training in computer programming and machine learning. In summary, a more systematic evaluation of cell segmentation methods is still needed to guide the practice of cell segmentation in real-world applications.

To address these limitations, we conducted a systematic evaluation of computational methods for cell segmentation, aiming to provide guidelines for applying and developing these methods in practical applications. The importance of this study is threefold. For researchers who are designing experiments, this study can help to decide a cost-effective strategy for generating imaging data. For researchers who have collected imaging data, this study can help identify the best-performing method and associated training data, depending on the available computational resources. For machine learning researchers, this study sheds light on potential opportunities that could lead to the development of more powerful cell segmentation methods in the future. In addition, we have also provided the pre-trained models for each cell segmentation method, tissue type, and imaging technology as a publicly available and freely accessible resource, significantly reducing the effort and computational resources required for training cell segmentation methods.

RESULTS

Overview

In this study, we evaluated the performance of 18 instance segmentation methods for both light microscopy and fluorescence staining images. These 18 methods can be divided into two categories. The first category comprises methods designed for biomedical images, including Cellpose[19], Mesmer[22], StarDist[20], RetinaMask[21], and FeatureNet[23]. The second category comprises methods designed for generic instance segmentation tasks, including Centermask2[24], Mask RCNN[25], ResNeSt[26], MS RCNN[27], Mask2former[28], Cascade Mask RCNN Seesaw[29], Swin Transformer (Swin-S and Swin-T)[30], SOLOv2[31], RF-Next[32], HRNet[33], Res2Net[34], and Segment Anything[35]. We include both Swin-S and Swin-T, two versions of the Swin Transformer with different model sizes, to explore how model complexity impacts segmentation performance. Segment Anything is based on a foundation model that does not require additional training or fine-tuning.

The entire evaluation dataset comprises more than 12,400 images and approximately 3 million cells, representing ten different tissue types and eight different cell types. We collected 7,022 images generated by multiplexed fluorescence imaging platforms from TissueNet[9,41]. Each image includes a nuclear channel (such as DAPI) and a membrane or cytoplasm channel (such as E-cadherin or Pan-Keratin). These images feature 1.3 million whole cells and 1.2 million cell nuclei from ten tissue types (Figure 1a). Additionally, we collected 5,387 images generated by phase-contrast microscopy from LIVECell[8,40]. These images are single-channel and contain 1,686,352 cells from eight cell types (Figure 1b). Both TissueNet and LIVECell provide annotated cell or nuclei masks for each image, which were used as training labels and as gold standards for evaluations.

Five types of segmentation tasks were evaluated (Figure 1c). In TissueNet, segmentation methods were trained either with nuclear channel images or dual-channel images for cell nuclei segmentation. Similarly, methods were trained either with whole cell channel images or dual-channel images for whole cell segmentation. For LIVECell, both model training and whole cell segmentation were performed using whole cell channel images. We assessed the performance of segmentation methods trained by aggregating images from all tissue types or cell types, as well as those trained with images from a single tissue type or cell type. The performance of each method was evaluated by comparing the predicted cell boundaries to the gold standard cell masks (Figure 1d, Methods). In accordance with the COCO evaluation metrics[22,42,43], we employed multiple quantitative metrics, including average precision (AP), mean average precision (mAP), and average recall (AR).

In addition to segmentation accuracy, we also assessed the scalability and usability of these methods. Scalability, measured by the running time of a method (Methods), reflects its time efficiency and the computational resources required. Usability, gauged by factors such as code maintenance, ease of use, and hardware support (Methods), indicates the ease with which the method can be implemented and executed in a new computing environment. These metrics provide valuable insights for practical application.

General-purpose methods with attention mechanism have the best performance

We systematically evaluated the accuracy, scalability, and usability of 18 cell segmentation methods and ranked them based on overall performance (Figure 2a, Supplementary Figure 1, Methods). Figure 2b shows examples of cell nuclei segmentation with dual-channel images for each method. While the top-performing methods produce results highly consistent with the gold standard, underperforming methods exhibit various issues. These include oversegmentation, undersegmentation, and incorrectly identifying non-cellular structures as cells. MS RCNN fails to generate reasonable cell boundaries due to the vanishing gradient

problem encountered during the training process.

Methods designed for general purposes outperform those developed specifically for biomedical images. Swin-S and Swin-T, both variants of the Swin Transformer[30] and intended for general purposes, demonstrate the best overall performance. In each of the five accuracy metrics, both Swin-S and Swin-T consistently rank among the top 5 or 6 methods, evidencing their reliable performance across various scenarios. The two models exhibit almost identical performances, indicating that model complexity has minimal impact on the results for the Swin Transformer. Centermask2 and ResNeSt, also designed for general purposes, achieve the highest accuracy in segmenting both nuclei and whole cells using dual-channel images in the TissueNet dataset, and in segmenting whole cells in the LIVECell dataset. However, their overall performance is negatively impacted by their lower scalability and usability. Among methods specifically designed for biomedical imaging, Cellpose has the best overall performance. It shows the highest accuracy in segmenting nuclei with single-channel images in the TissueNet dataset and performs strongly whenever nuclei images are available. However, its performance significantly declines when nuclei images are absent. Other methods tailored for biomedical images do not exhibit strong performance.

Methods exhibiting the highest overall accuracy, such as Centermask2, ResNeSt, Swin-S, and Swin-T, all incorporate the attention mechanism[44]. While Mask2former and Segment Anything also use the attention mechanism, they underperform in this study. Notably, Segment Anything, a pre-trained foundation model not fine-tuned on our training images, may underperform due to this reason. Other methods employing different architectures, like Cellpose with U-Net[45] and Cascade Mask RCNN seesaw with ResNet[46] and FPN[47], also demonstrate competitive performance.

Dual-channel images greatly enhance whole cell segmentation over single-channel images

Figure 2a compares the performance of segmentation methods trained with dual-channel versus single-channel images. Dual-channel images significantly enhance whole cell segmentation for most methods. For instance, Cellpose and Centermask2's accuracy rose from 0.17 to 0.7 (312% improvement) and 0.3 to 0.79 (163% improvement), respectively. In contrast, the performance boost from dual-channel images is minimal for cell nuclei segmentation. Only a few methods like Centermask2, ResNeSt, and StarDist see substantial gains with over 0.1 accuracy increase. Methods such as Swin-S, SOLOv2, and Res2Net show similar performance with both image types. However, for some, including Mask2former and Segment Anything, performance notably declines with dual-channel images. These findings indicate varying capabilities among methods in integrating and utilizing multi-channel images. Additionally, while cell nuclei images alone suffice for nuclei segmentation, whole cell segmentation benefits from additional data in dual-channel images.

The performance gain from using dual-channel over single-channel images for each segmentation method varies across different tissues (Supplementary Figure 2). For instance, in lung tissue, Centermask2's accuracy for cell nuclei segmentation jumps from 0.308 to 0.655 (113% improvement), while in esophagus, the increase is more modest, from 0.691 to 0.759 (9.8% improvement). These findings are crucial for experimental design. In tissues like the esophagus, using single-channel images for cell nuclei segmentation yields results comparable to dual-channel images, potentially reducing experiment costs. However, for whole cell segmentation and for cell nuclei segmentation in tissues like the lung, employing dual-channel images is advised for significantly better segmentation outcomes.

Segmentation performance is influenced by training sample size and cell morphology

We explored the influence of various factors on cell segmentation performance across five tasks, using training images from all tissue or cell types. Figure 3a shows the mAP of different methods when tested on various tissue or cell types. Generally, there is a consistent trend in method performance across these tissues or cell types. For instance, lymph node metastasis, pancreas, and esophagus consistently yield reliable segmentation results with most methods. In contrast, lung and spleen are often associated with poor segmentation, primarily due to the extremely limited number of images in the training data (Supplementary Figure 3). These observations suggest that there are some common factors influencing segmentation performance across different methods.

To identify common factors influencing cell segmentation, we excluded lung and spleen from the TissueNet dataset and focused on tissue and cell types with sufficiently large numbers of training images. We employed a multiple linear regression model, considering the method's mAP as the dependent variable, and the number of training images, cell elongation, and cell size as independent variables (Methods). Cell convexity was excluded due to its minimal variation across tissues (Supplementary Figure 4). For most methods and segmentation tasks, there is a positive correlation between segmentation performance and cell elongation, cell size, and the number of tissue-specific training images (Figure 3b). The correlation with the number of training images is expected, as images from the same tissue typically bear more resemblance to each other than to those from different tissues. The positive association with cell elongation and size indicates that segmentation algorithms are most effective for larger or rounder cells, such as those in lymph node metastasis. Conversely, smaller or irregularly shaped cells, like those in lymph nodes and the colon (Figure 3c, Supplementary Figure 5), pose challenges. This could be because smaller cells lack distinctive features for differentiation from the background, and irregularly shaped cells are less represented in the training data. Enhancing segmentation for such cells might be achieved by increasing their representation in the training set or integrating a model specifically trained on these cell types into the full model using an ensemble approach.

Choice of training data impacts cell segmentation performance

We subsequently assessed how the performance of Swin-S, the top-performing method, is affected when trained with different tissue or cell types. Figures 4a-b and Supplementary Figures 6-9 illustrate Swin-S's performance in the TissueNet dataset, trained individually with different tissue types or with all tissues combined. Notably, the segmentation performance is poorest when Swin-S is trained solely with lung or spleen images, likely due to their very limited number of training images. For tissues with adequate image quantities, segmentation performance varies with the training tissue. The best results are typically achieved when the same tissue is used for both training and segmentation. When training and segmentation tissues differ, some tissues demonstrate better transferability. For instance, Swin-S trained on breast tissue images consistently shows reliable nuclei segmentation across various tissues, whereas training with tonsil images results in subpar performance in tissues like the esophagus. Furthermore, Swin-S trained on all tissues generally outperforms its counterpart trained on a single tissue, although the performance improvement is slight when the training and segmentation are on the same tissue. Figure 4c and Supplementary Figure 10 illustrate Swin-S's performance when trained with individual versus all cell types in LIVECell, leading to conclusions similar to those from the TissueNet analysis. For instance, cell types like MCF-7 and BT-474 demonstrate relatively high transferability across different cell types.

These results indicate that including images from diverse tissues and cell types in the training dataset is advantageous when

sufficient computational resources are available. In scenarios with limited computational resources, it's essential to determine the availability of training images from the same tissue or cell type as the target tissue for segmentation. Using such specific images for training often yields performance nearly equivalent to that achieved with a large, varied dataset. If not available, training should utilize images from tissues and cell types with high transferability, like breast tissue and MCF-7 cells. Figure 4d summarizes this practical strategy for selecting training data.

Generalization of segmentation models across image modalities

In real applications, the images employed for model training and segmentation often come from different modalities. For instance, *in situ* spatial transcriptomics data from platforms like Xenium typically provide only cell nuclei images, yet whole cell segmentation is necessary to fully capture a cell's gene expression profile. Similarly, there are scenarios where only cell nuclei images are available for training, but segmentation must be conducted on whole cell images. These situations necessitate a segmentation method's ability to generalize across multiple image modalities.

Figure 5a and Supplementary Figure 11a illustrate the performance of segmentation methods when training and segmentation use the same or different image modalities. The performance of most methods drops significantly when trained on whole cell images and applied to cell nuclei images, compared to training and segmenting on cell nuclei images. However, Cascade Mask RCNN seesaw and RF-Next maintain nearly identical performance in both scenarios, demonstrating strong generalizability across image modalities. By contrast, all methods completely fail when trained on cell nuclei images and tested on whole cell images. These findings indicate that most segmentation methods struggle with generalizability across different image modalities.

We then explored a strategy of segmenting images using the same modality as the training data, followed by morphologically dilating cell nuclei segmentation to approximate whole cell segmentation, or eroding whole cell segmentation for cell nuclei segmentation (Methods). This approach is currently used by 10x Xenium to provide an approximate whole cell segmentation, where the cell nuclei mask is expanded by 15 μm or until reaching another cell boundary. Figure 5b and Supplementary Figure 11b demonstrate that expanding cell nuclei segmentation by varying pixel counts generally enhances whole cell segmentation performance. For example, in epidermis tissue, Swin-S shows a dramatic increase from 0.341 to 0.598 (75% improvement) with expansion. However, the optimal expansion degree varies across tissues. In ResNeSt, tonsil segmentation peaks at a 6-pixel expansion, whereas colon segmentation deteriorates with any nuclei boundary expansion. Therefore, a fixed-pixel expansion, as implemented by 10x Xenium, may not be optimal for all tissues. Figure 5c and Supplementary Figure 11c present the results of eroding whole cell segmentation for cell nuclei segmentation. The performance gains are more modest, with most cases showing optimal results with a one-pixel erosion.

SegGal: a gallery of pre-trained cell segmentation models

As shown in this study (Figure 4) and in our previous work[7], cell segmentation performance is significantly influenced by the training data. Pre-trained models, developed using a vast array of images from various tissues and cell types, can greatly enhance cell segmentation. However, training these models demands considerable computational resources and the effort to collect a large number of images. Despite their potential benefits, such comprehensive pre-trained models are often not readily available. While generic segmentation methods typically lack pre-trained models specialized for cell images, biomedical image segmentation methods like Cellpose only offer pre-trained models trained on a small numer of images.

To address this challenge, we have developed SegGal, a gallery of pre-trained cell segmentation models, leveraging the numerous models trained in this study. SegGal includes 138 pre-trained models across 18 segmentation methods, catering to five different segmentation tasks. The training process utilized 450 GPUs and required 18,000GB of memory, spanning approximately 60 days on the computing clusters of three universities. This effort is equivalent to over 70 years of training with a single GPU. These pre-trained models are readily available for download from the SegGal website, allowing users to apply them directly to segmentation tasks. This not only significantly reduces the computational load for model training but also streamlines the cell segmentation process.

DISCUSSION

In this study, we systematically evaluated 18 segmentation methods for their effectiveness in segmenting cell nuclei and whole cells using multiplexed fluorescence and phase-contrast microscopy images from the TissueNet and LIVECell databases. These methods, encompassing both general-purpose and biomedical image-focused approaches, were assessed based on accuracy, usability, and scalability. We investigated various factors influencing segmentation performance, such as the use of single or dual-channel images, training sample size, choice of training data, and cell morphology. Additionally, we evaluated the methods' ability to generalize across different image modalities. Lastly, we developed Seggal, an online resource providing access to a vast collection of pre-trained cell segmentation models.

Our findings contribute to the study of cell segmentation in three key areas. For researchers planning to generate imaging data, our study indicates that cell nuclei images are crucial for both cell nuclei and whole cell segmentation, with whole cell images proving beneficial in certain scenarios. For those developing cell segmentation methods, our findings highlight the attention mechanism, utilized by several top-performing methods, as a promising architecture worthy of further exploration. Additionally, our study offers a systematic guideline for researchers seeking to apply these methods in cell segmentation. Figure 6 provides a comprehensive list of cell segmentation methods suitable for various scenarios. Through the Seggal online resource, users gain direct access to a vast array of pre-trained models, significantly easing the burden of model training across different scenarios.

As indicated by this study, segmentation performance improves when training data includes a large variety of images from different tissues and cell types. Although 'Segment Anything,' the sole foundation model evaluated in our study, did not perform well, we hypothesize that such a model could have significant potential if trained with massive, diverse datasets of cell images. TissueNet's adoption of a human-in-the-loop approach demonstrates a viable method for generating extensive training datasets at a relatively low cost. Such datasets could pave the way for developing a foundational model broadly applicable to cell segmentation and potentially other biomedical image segmentation tasks.

METHODS

Datasets

TissueNet[22] dataset V1.1, available on the TissueNet website[41], comprises 7,022 staining images. These are distributed as 2,580 images in the training set, 3,118 in the validation set, and 1,324 in the testing set. The images span ten tissue types of breast, colon, epidermis, esophagus, lung, lymph node, pancreas, spleen, tonsil, and lymph node metastasis, and originate from three species of human, mouse, and macaque. Each image features a nuclear staining channel and a membrane or cytoplasm

whole-cell staining channel, along with annotations for both nuclei and whole cells, derived from an iterative human-in-the-loop process[22]. The training set images have a resolution of 512×512 pixels, while those in the validation and testing sets are 256×256 pixels each.

LIVECell[48] dataset was obtained from the LIVECell website[40]. It encompasses 5,387 microscopy images, divided into 3,253 for training, 570 for validation, and 1,564 for testing. These images represent eight distinct cell types with unique morphological features, including A172, BT-474, BV-2, Huh7, MCF7, SH-SY5Y, SkBr3, and SK-OV-3. Each image, with a resolution of 704×520 pixels, includes manually labeled and expert-validated cell annotations.

Image preprocessing

All images in TissueNet and LIVECell datasets were preprocessed following a common practice in the field[19,22]. Specifically, each image along with the corresponding annotations were cropped to sliding windows. The size of each window is 256×256 and the step size is 75% of window size (192 pixels). The sliding windows scan across the image, from left to right and top to bottom, at each step making a local crop. After reaching the end of a row, the window moves down by the step size and continues scanning until the entire image has been covered, ensuring comprehensive coverage and increasing the likelihood of accurate detection. Pixel values were then normalized to account for exceptionally bright, isolated pixels, variations in staining intensity, and signal-to-noise ratios across tissue types and imaging platforms[49]. Specifically, we first compute the 0.1% (Min_p) and 99.9% (Max_p) intensity percentiles of an image (V) using *numpy.percentile* and then the normalized image ($\hat{V}$) is obtained by applying Max-Min normalization, expressed as $\hat{V} = (V - Min_p)/(Max_p - Min_p)$, thereby ensuring effective outlier exclusion in the process. Furthermore, we employed Contrast Limited Adaptive Histogram Equalization (CLAHE)[50] to balance the dynamic range within each image.

Segmentation methods

Unless otherwise specified in the following, all methods were trained and applied with the default parameters. Each method was adjusted to fit the input image dimensions, including height, width, and channel number. These methods, including Swin-S, Swin-T, Cascade Mask RCNN seesaw, SOLOv2, Res2Net, RF-Next, HRNet, Mask2former, Mask RCNN, and MS RCNN, were performed on MMDetection platform (version 2.28.2), an open-source object detection toolbox.

Cellpose[19] employs the general U-Net[45] architecture, which involves multiple downsampling steps of convolutional features, followed by a symmetrical upsampling process. Instead of feature concatenation, Cellpose uses direct summation between the upsampling and downsampling phases to reduce the number of parameters. Additionally, its standard U-Net building blocks are replaced with residual blocks[46] to enhance performance. Cellpose (version 1.0.2) was downloaded from `https://github.com/MouseLand/cellpose` and trained on a single GPU.

Mesmer[22] utilizes a ResNet-50[46] backbone model for feature pyramid extraction in image analysis. These extracted features are subsequently transformed into pyramid structures via a Feature Pyramid Network (FPN)[47]. The pyramid features are then employed to generate predicted cell instance masks, matching the size of the input image, through semantic segmentation heads. Mesmer (version 0.12.9) was downloaded from `https://github.com/vanvalenlab/deepcell-tf`. For training,

Mesmer was configured on 4 GPUs with a learning rate of 0.0004, a batch size of 8, and a total of 200 epochs.

StarDist[20] employs star-convex polygons for cell instance localization, offering superior shape representation compared to bounding boxes without the need for further shape refinement. It is built upon the U-Net[45] architecture. To prevent feature competition between the two subsequent output layers, StarDist integrates an additional 3×3 convolutional layer with 128 channels, accompanied by ReLU activations. Specifically, it utilizes a single-channel convolutional layer with sigmoid activation for object probability output, and for polygon distance estimation, the number of output layers aligns with the radial directions without an extra activation function. StarDist (version 0.8.3) was downloaded from `https://github.com/stardist/stardist`. StarDist was trained on a single GPU for 100 epochs.

RetinaMask[21], an advanced version of RetinaNet[51], serves as a unified, one-stage object detection system. It utilizes a ResNet-50 backbone network to compute a comprehensive convolutional feature map. This backbone connects to two subnetworks: one for convolutional object classification and another for bounding box regression, based on the extracted feature map. The key advancement of RetinaMask over RetinaNet is the introduction of instance mask prediction during training. As a result, RetinaMask outputs not only bounding boxes and classifications but also precise cell instance masks. RetinaMask (version 0.1.1) was downloaded from `https://github.com/vanvalenlab/deepcell-retinamask`. For training, RetinaMask was configured on 4 GPUs with a learning rate of 0.0005 over 100 epochs.

FeatureNet[23] consists of an encoder for generating a low-dimensional feature representation of the input image, utilizing convolutional filters, activation functions, and max-pooling operations. These features are then input into a classifier, based on a fully connected network, which assigns probability scores to three classes: boundary, interior, and background. These probabilities are used to create the final cell instance masks. FeatureNet (version 0.12.9) was downloaded from `https://github.com/vanvalenlab/deepcell-tf`. For training, FeatureNet was configured on 4 GPUs with a learning rate of 0.001, a batch size of 16, and 25 epochs.

Centermask2[24] comprises three key components. The first is the VoVNetV2, an efficient backbone network for feature extraction. The second, FCOS[52], is an anchor-free, proposal-free object detection network that directly predicts bounding box coordinates through per-pixel analysis. The final component, the Spatial Attention-Guided Mask (SAG-Mask), utilizes outputs from the first two components for precise cell instance mask prediction. Centermask2 was downloaded from `https://github.com/youngwanLEE/centermask2`. Centermask2 was configured on 4 GPUs with 100,000 iterations and a batch size of 32.

Mask RCNN[25] follows the spirit of Faster R-CNN architecture[53] with the same two-stage procedure. The first stage generates proposals about the regions where there is an object (Region Proposal Network, RPN[53]), which is identical first stage with Faster R-CNN. In the second stage, in parallel to the branch predicting the class and bounding box, Mask RCNN adds an additional segmentation branch to output a binary mask for each Region of Interest (ROI). Mask RCNN was performed on 4 GPUs with 100 epochs and 24 batch sizes.

MS RCNN (Mask Scoring R-CNN)[27] improves upon the instance segmentation capabilities of Mask RCNN. Unlike Mask RCNN, which uses the segmentation branch's output as the final mask prediction, MS RCNN adds a mask scoring branch. This branch calculates a mask *IoU* (Intersection over Union) score, assessing the overlap between the predicted and ground truth binary masks. This score provides a more accurate evaluation of instance segmentation quality without adding significant complexity or computational burden. MS RCNN was trained on 4 GPUs with a configuration of 50 epochs and a batch size of 48.

ResNeSt[26] introduces a multi-branch architecture that employs channel-wise attention across its branches, combining the strengths of feature-map attention and multi-path representation. Its key innovation, the Split-Attention (SA) module, acts as a modular, versatile block, easily substituting the conventional residual block. This module fosters more diverse representations through cross-feature interactions. Such diversity enhances segmentation performance without increasing model complexity or reducing computational efficiency, thereby offering a scalable solution for various tasks. ResNeSt (version 0.1.1) was downloaded from `https://github.com/chongruo/detectron2-ResNeSt`. ResNeSt was configured on 4 GPUs with a ResNet FPN backbone, performing 50,000 iterations and using a batch size of 4.

Mask2former[28], a universal image segmentation architecture, evolves from the Maskformer meta-architecture[54] and includes a ResNet-50 backbone for feature extraction, a pixel decoder, and a Transformer decoder. Its key advancement is the adoption of a Transformer decoder that applies masked attention, focusing on localized features around predicted segments, rather than using conventional cross-attention. These segments, derived from the pixel decoder, enable the refined output of object instances. Mask2former was configured on 4 GPUs for 50 epochs and a batch size of 36.

Cascade Mask RCNN seesaw[29] enhances Cascade Mask RCNN by introducing an innovative seesaw loss function to address the class imbalance issue common in object detection and instance segmentation tasks. While traditional methods like Focal Loss[51] provide some relief, they may fall short in extreme class imbalance situations. Seesaw loss dynamically adjusts loss weights for each class, based on the class sample size in each mini-batch. It employs a seesaw mechanism that increases the weight for minority classes and decreases it for majority classes, ensuring more balanced learning. Additionally, it gives more weight to harder samples within each class to further equalize the learning process. Cascade Mask RCNN seesaw was configured on 4 GPUs for 50 epochs and a batch size of 8.

Swin Transformer[30] merges the strengths of convolutional neural networks (CNNs) and transformers, offering a scalable and efficient approach for various visual tasks. Its hierarchical structure is a key innovation, utilizing shifted window-based self-attention mechanisms to process local information within small, discrete windows. These windows are progressively merged and resized across stages, enabling the model to effectively capture both local and global contexts. The primary distinction between Swin-T and Swin-S lies in their size and computational complexity, with Swin-S being twice as large as Swin-T. Both variants were trained on 4 GPUs for 50 epochs and a batch size of 8.

SOLOv2[31], an evolution of the original SOLO[55], adopts a refined approach that focuses on direct per-pixel predictions instead

of conventional bounding-box techniques. At its core, SOLOv2 features a fully convolutional network structure, category-specific kernel predictions, and the innovative Matrix Non-Maximum Suppression (Matrix NMS) for refining overlapping instances. Initially, it utilizes the same ResNet-50 backbone and FPN as SOLO for feature extraction, then integrates a category-specific kernel prediction network to infer all instance masks. Matrix NMS, notably faster and more accurate than traditional NMS[56], enhances the accuracy and efficiency of the model. SOLOv2 was trained on 4 GPUs for 60 epochs with a batch size of 8.

RF-Next[32] primarily follows the Multi-Stage Temporal Convolutional Network (MS-TCN)[57] approach, introducing a global-to-local search method that includes two innovative components: a genetic-based global search algorithm for generating competitive receptive field combinations, and an expectation-guided iterative local search for fine-tuning these combinations. Specifically, RF-Next uses Mask RCNN as its instance segmentation foundation and integrates these components into convolutional layers with kernels larger than one, optimizing segmentation results. RF-Next was configured on 4 GPUs for 200 epochs with a batch size of 8.

HRNet[33] introduces an architecture that features parallel multi-resolution convolutions (PMRC), repeated multi-resolution fusions (RMRF), and a representation head maintaining high-resolution representations throughout. Diverging from traditional models that downsample and then upsample, HRNet starts with a high-resolution subnetwork and progressively integrates additional high-to-low resolution subnetworks in stages. It continuously fuses these multi-resolution subnetworks in parallel, enabling dynamic feature interplay. HRNet's core design focuses on iterative multi-resolution fusions, constantly exchanging and integrating information across these parallel subnetworks to capture both fine and coarse features in the input data comprehensively. HRNet was configured on 4 GPUs for 200 epochs with a batch size of 24.

Res2Net[34] introduces a modification to the ResNet architecture with a granular multi-scale design. Unlike traditional ResNet models, Res2Net splits input features in each residual block into subsets. Each subset processes different scales using varying dilation rates, functioning similarly to parallel sub-ResNets. These multi-scale feature maps are then efficiently integrated through convolution, enabling effective information sharing between scales. The key strength of Res2Net lies in its hierarchical block structure, which facilitates enhanced feature propagation and efficient exploitation of multi-scale features. Res2Net was configured on 4 GPUs for 20 epochs with a batch size of 8.

Segment Anything[35] introduces the Segment Anything Model (SAM) for promptable segmentation, incorporating three key components: a Masked AutoEncoders (MAE)[58]-based image encoder adapted for high-resolution inputs, a flexible prompt encoder (PE) capable of processing various prompts, and a fast mask decoder (ME). SAM utilizes a pre-trained Vision Transformer (ViT[59]) as its image encoder. The prompt encoder can handle both sparse prompts (interpreted via positional encodings[60] and combined with specialized embeddings) and dense prompts (integrated with the encoded image). The mask decoder then amalgamates the embeddings from both components to generate segmentation masks using a modified Transformer decoder block[44], followed by a dynamic mask prediction head, which outputs the mask foreground probability for each pixel. Segment Anything, downloaded from https://github.com/facebookresearch/segment-anything, was employed for segmentation, utilizing its default settings and the pre-trained ViT-H SAM model. Consequently, the process

did not involve any GPU-based training.

Execution and Parameter settings

Each method was independently executed on a dataset for a specific segmentation task. For each task, we allocated 200G of memory and either a single GPU or multiple GPUs of the same type, depending on the method's support for multi-GPU training. To ensure fair comparison across all methods, we adhered to the parameter settings specified in their original publications.

Segmentation benchmarks

We assessed segmentation performance in this study using the COCO evaluation metrics[22,42,43] from `pycocotools` package (version 2.0.6), a standard evaluation method in instance segmentation. To ensure all cell instances per image were considered, we increased the parameter for maximum detections per image to 3000, from the default value of 100. The details of the evaluation metrics are described below:

Intersection over Union (IoU)

For the oth image in the test set with resolution of $H \times W$ pixels, let $\{S^p_{k,o} \in \{0,1\}^{H\times W}\}_{k=1}^{K}$ be the predicted instances of K cells, and let $\{S^g_{k',o} \in \{0,1\}^{H\times W}\}_{k'=1}^{K'}$ be the ground truth annotations of K' cells. The Intersection over Union (IoU) score of the kth predicted cell instance is computed by:

$$IoU^p_{k,o} = \max\left(\left\{\frac{S^p_{k,o} \cap S^g_{k',o}}{S^p_{k,o} \cup S^g_{k',o}}\right\}_{k'=1}^{K'}\right). \tag{1}$$

Likewise, the IoU score of the k'th ground truth cell annotation is computed by:

$$IoU^g_{k',o} = \max\left(\left\{\frac{S^p_{k,o} \cap S^g_{k',o}}{S^p_{k,o} \cup S^g_{k',o}}\right\}_{k=1}^{K}\right). \tag{2}$$

Precision and Recall

Let t be a threshold taking values from $\{t \mid t = 0.5 + 0.05i, 0 \le i \le 9\}$. For a given threshold t, a predicted cell instance k is considered a true positive (TP) if $IoU^p_{k,o} \ge t$, and a false positive (FP) if $IoU^p_{k,o} < t$. A ground truth cell annotation k' is considered a false negative (FN) if $IoU^g_{k',o} < t$.

For the oth image and with threshold t, let $TP_{t,o}$, $FP_{t,o}$, and $FN_{t,o}$ be the total number of true positive, false positives and false negatives, respectively. The precision $P_{t,o}$ is defined as $P_{t,o} = \frac{TP_{t,o}}{TP_{t,o}+FP_{t,o}}$. The recall $R_{t,o}$ is defined as $R_{t,o} = \frac{TP_{t,o}}{TP_{t,o}+FN_{t,o}}$.

Average Precision and Average Recall

Across all O images in the testing set, given precision $\{P_{t,o}\}_{o=1}^{O}$ and the recall $\{R_{t,o}\}_{o=1}^{O}$, a step-wise precision-recall curve is built, expressed as $P_{t,o} = \Psi(R_{t,o})$ and then to ensure the curve is monotonically decreasing, an interpolated curve $\hat{P}_{t,o} = \hat{\Psi}(R_{t,o})$ is obtained by calculating the $\hat{P}_{t,o-1} = \max_{R_{t,o} \ge R_{t,o-1}} (\Psi(R_{t,o}), \Psi(R_{t,o-1}))$. Finally, the average precision with threshold t (AP_t) is computed by:

$$AP_t = \sum_{o=2}^{O} (R_{t,o} - R_{t,o-1})\hat{\Psi}(R_{t,o}). \tag{3}$$

The average recall with threshold t (AR_t) is computed by:

$$AR_t = \frac{1}{O}\sum_{o=1}^{O} R_{t,o}. \tag{4}$$

Mean averaged precision (mAP) is calculated as the average of AP_t across all possible values of t.

Cell morphology

Cell convexity. Let r_{cell} be the perimeter of a cell's boundary, and let r_{convex} be the perimeter of the cell's convex hull, the smallest possible convex shape that completely contains the cell region. The cell convexity of the cell is defined as $r_{\text{convex}}/r_{\text{cell}}$.

Cell elongation. Let B be a set of two-dimensional points representing the boundary of a cell. We compute e_a and e_b, the two eigenvalues of the covariance matrix of B, where $e_a > e_b$. Cell elongation is computed as e_b/e_a.

Cell size. The size of a cell instance is defined as the number of pixels contained within the boundary of the cell.

Morphological Operations

The `binary_dilation` and `binary_erosion` functions from the `scipy` (version 1.9.2) multidimensional image processing (`ndimage`) package were used for cell nuclei mask dilation and whole cell mask erosion, respectively.

Scalability

For each dataset evaluated in this study, we recorded the running time T of a segmentation method as the time taken to finish model training after all packages were loaded. Different segmentation methods have different choices of batch size and number of epochs or iterations, which may affect their running time. To ensure a fair comparison across methods, we calculated standardized running time as $T_s = T \times \frac{B}{8} \times \frac{50}{E}$ for methods using epochs as a parameter, and $T_s = T \times \frac{B}{8} \times \frac{50N}{I \times B}$ for methods using iteration as a parameter. Here B, E, I, and N represent the batch size, number of epochs, number of iterations, and number of images training samples, respectively.

Usability

We evaluated the usability of each method using three metrics, each manually scored on a scale from 1 to 10, with higher scores indicating better performance. The scores for these three metrics are shown in Supplementary Table 1.

Code Maintenance assesses the ease of updating and maintaining a method's programming code. This metric takes into account the clarity and readability of the code, adherence to good software engineering practices (such as modular design, encapsulation, version control system usage, and comprehensive commenting), and the availability of supporting documentation.

Ease of Use assesses the simplicity with which end users and developers can utilize and interact with the method. Key considerations include the ease of training the model, adjusting parameters, interpreting outputs, and resolving any encountered issues.

Hardware Support assesses a method's compatibility with various GPU architectures and its ability to fully leverage specific hardware features, such as GPUs' parallel processing capabilities. Additionally, this criterion considers the method's adaptability to different hardware configurations and its capacity to efficiently scale according to available resources.

Ranking Scheme

The methods are ranked by their overall performance. The overall performance is calculated as the average of seven metrics: the accuracy in TissueNet nuclei segmentation with dual-channel images, the accuracy in TissueNet nuclei segmentation with nuclei images, the accuracy in TissueNet whole cell segmentation with dual-channel images, the accuracy in TissueNet whole cell segmentation with whole cell images, the accuracy in LIVECell whole cell segmentation, averaged usability score, and averaged scalability score.

Each of the five segmentation accuracy is calculated as the average of five metrics of that task: mAP, $AP_{0.5}$, $AP_{0.75}$, $AR_{0.5}$, and $AR_{0.75}$.

The averaged usability score is calculated as the average of normalized scores of code maintenance, ease of use, and hardware support. To obtain the normalized score of code maintenance, the raw scores of code maintenance across 18 segmentation methods were first scaled to have a mean of 0 and a standard deviation of 1. The normalized score was then obtained by mapping the scaled scores to a range between 0 and 1 using the cumulative density function of a standard normal distribution. Normalized scores of ease of use and hardware support were obtained similarly.

The averaged scalability score is calculated as one minus the average of normalized running time of TissueNet nuclei segmentation, TissueNet whole cell segmentation, and LIVECell segmentation. Running time of TissueNet nuclei segmentation is calculated as the average running time of TissueNet nuclei segmentation with nuclei and dual-channel images. Similarly, running time of TissueNet whole cell segmentation is calculated as the average running time of TissueNet whole cell segmentation with whole cell and dual-channel images. The three raw running time were converted to scaled and then normalized running time using the same normalization method for calculating normalized usability scores.

DATA and CODE AVAILABILITY

The TissueNet dataset[41] is available on the website: https://datasets.deepcell.org/data.

The LIVECell dataset[40] is available on the website: https://sartorius-research.github.io/LIVECell/.

The preprocessing TissueNet datasets are available on Google Drive: https://drive.google.com/drive/folders/1dUtqhvkF-M7nSwtxUpY0QmgHIgA4pinc?usp=drive_link.

The preprocessing LIVECell datasets are available on Google Drive: https://drive.google.com/drive/folders/1mJayXI2W9DLL17fsD3j2AcFySebnsoza?usp=drive_link.

SegGal website is available on the website: https://boomstarcuc.github.io/SegGal/.

All methods used in this study are integrated into the GitHub: https://github.com/BoomStarcuc/Cell-Segmentation-Benchmark with License MIT.

ACKNOWLEDGEMENTS

Z.J. was supported by the National Institutes of Health under Award Number U54AG075936 and by the Whitehead Scholars Program at Duke University School of Medicine.

AUTHOR CONTRIBUTIONS

Z.J. conceived the study. Y.W., J.Z., H.X., and C.H. performed the analysis. J.Z., Z.T., D.Zhao, D.Zhou, G.T., D.L., and Z.J. provided computational resource and advised the analysis. Y.W., J.Z., H.X., D.L, and Z.J. wrote the manuscript.

COMPETING INTERESTS

None declared.

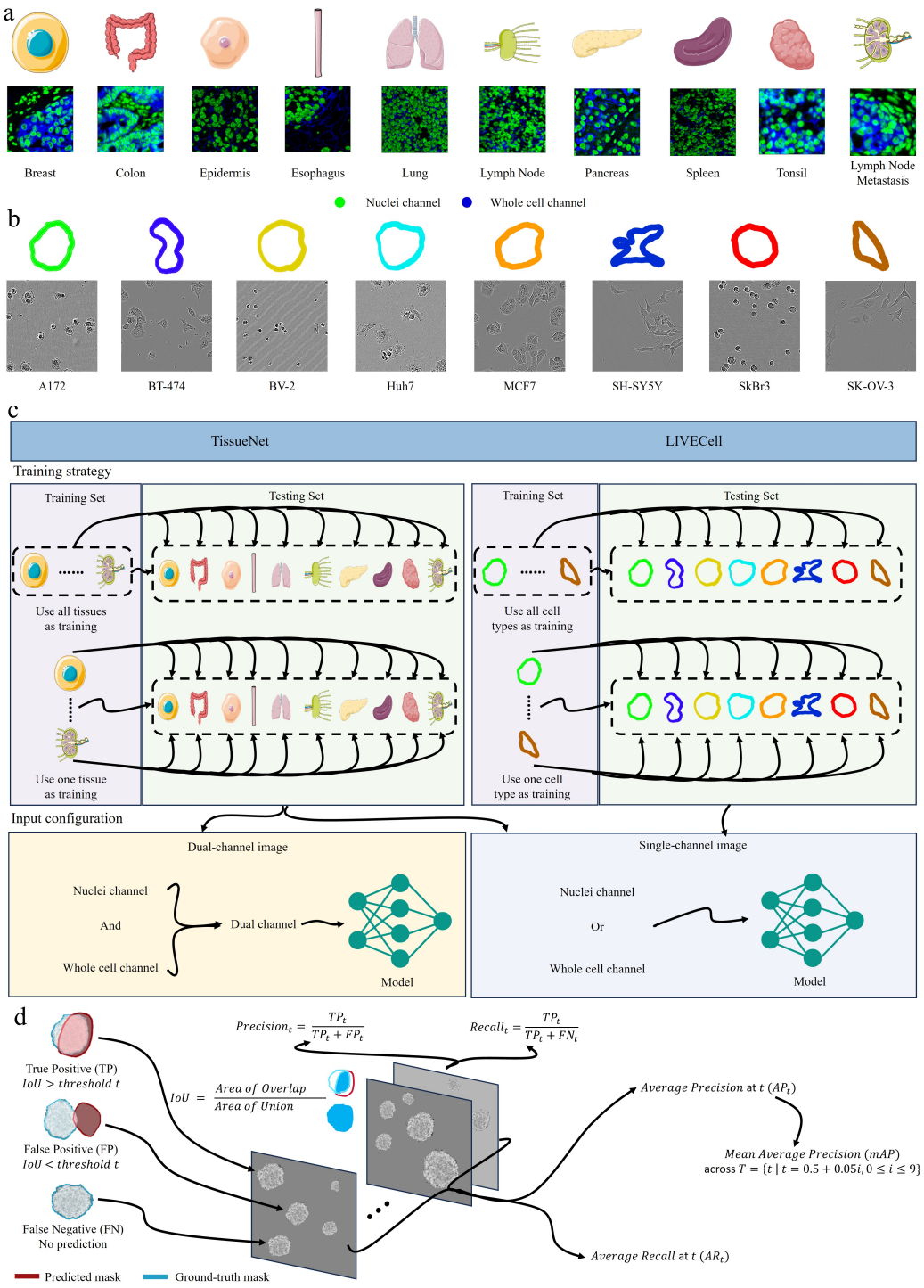

Figure 1. a, TissueNet dataset with ten tissue types. The image channel marked in green represents the nuclei channel, and the channel marked in blue represents the whole cell channel. **b**, LIVECell dataset with eight cell types. Only whole cell images are provided. **c**, The benchmarking overview for cell segmentation. This benchmarking study is conducted on two datasets, TissueNet and LIVECell, with two training strategies. TissueNet employs two input configurations due to nuclei and whole cell available while LIVECell only involves one input configuration due to only whole cell provided. **d**, Cell segmentation benchmarks with mAP, AP_t, and AR_t. The segmentation benchmarks strictly follow the COCO evaluation metrics[22,42,43], which is the standard method of evaluation in the field of instance segmentation, to comprehensively evaluate cell segmentation accuracy.

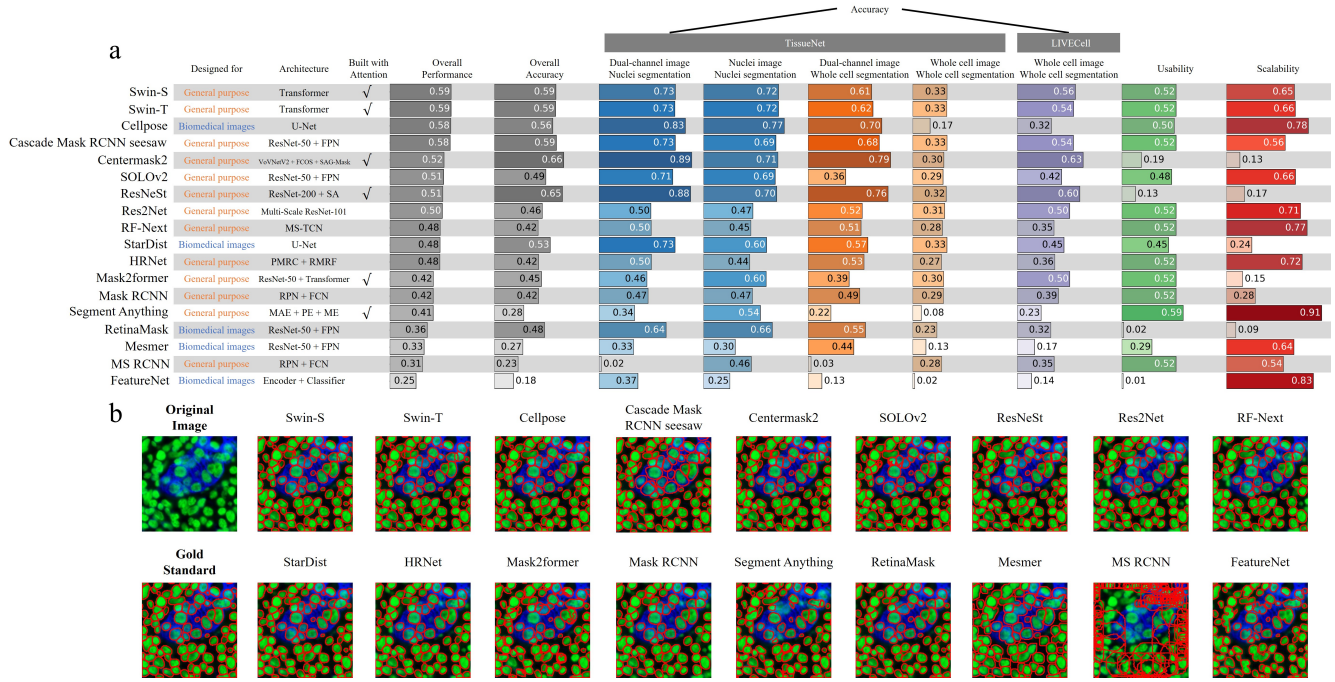

Figure 2. a, A comprehensive evaluation of methods involving segmentation accuracy, usability, and scalability. Segmentation accuracy is assessed on two datasets encompassing five segmentation tasks: nuclei segmentation with dual-channel images, nuclei segmentation with nuclei images, whole cell segmentation with dual-channel images, whole cell segmentation with whole cell images in TissueNet and whole cell segmentation with whole cell images in LIVECell. Usability and scalability metrics shed light on the practicality and training time efficiency of each method, respectively. Overall performance and overall accuracy are obtained by computing the arithmetic mean of the seven experiments and five experiments, respectively. Each row is a method, and each column is an experiment. All methods are sorted in descending order based on the overall performance. **b**, Visual results of lymph node metastasis in nuclei segmentation with dual-channel images.

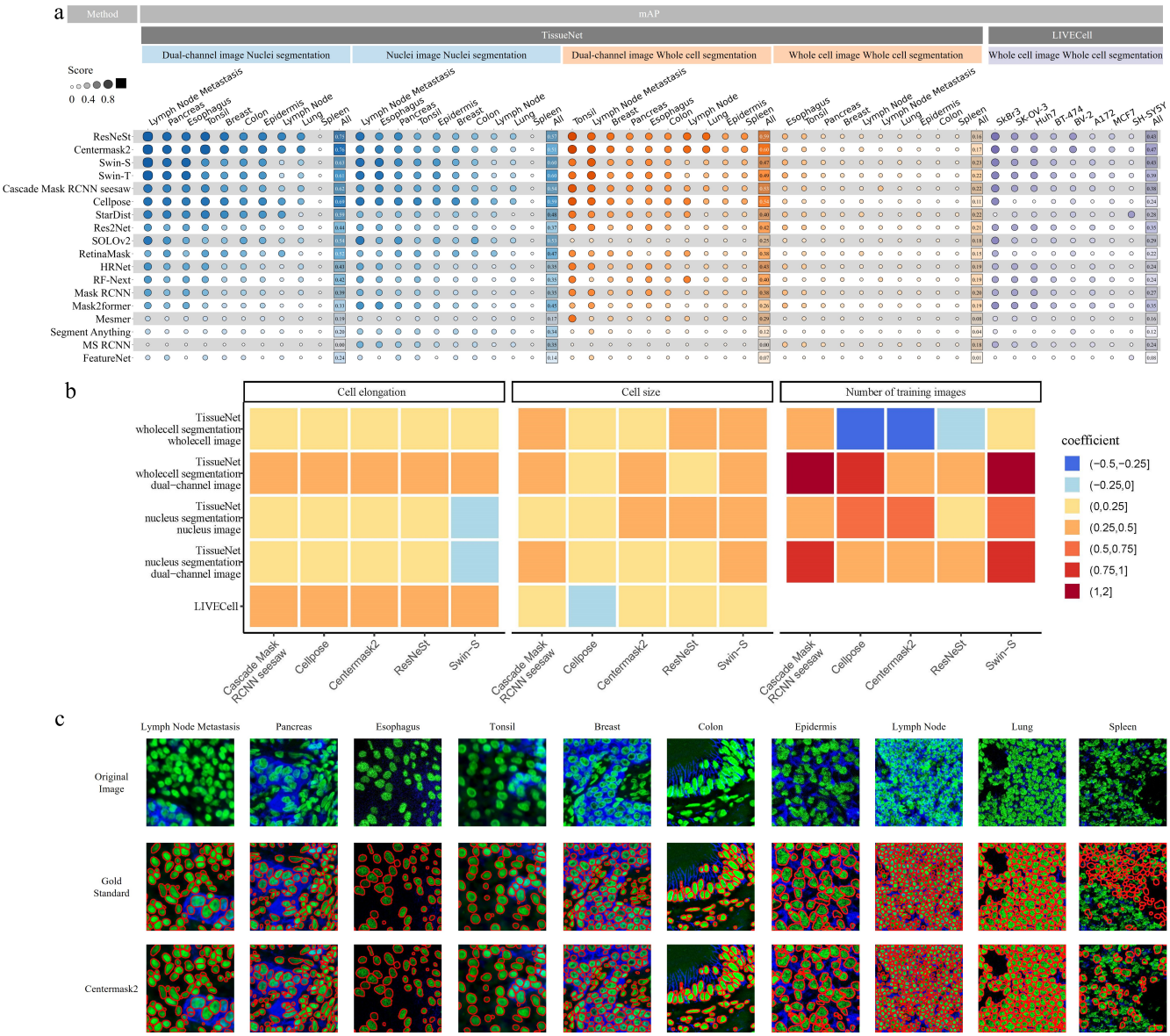

Figure 3. a, Evaluation results from training on all tissue or cell types and testing either on *All* types or on each tissue type or cell type. These results do not consider usability and scalability. Each row is a method, and each column is a type. All methods are ranked in descending order based on the arithmetic mean of the *All* scores across five segmentation tasks. Within per task, the tissue or cell types are ranked in descending order based on the arithmetic mean for each type across all methods. **b**, Correlation between segmentation performance and the number of training images, cell elongation, and cell sizes. **c**, Visual results of all tissue types for Centermask2 in nuclei segmentation with dual-channel images.

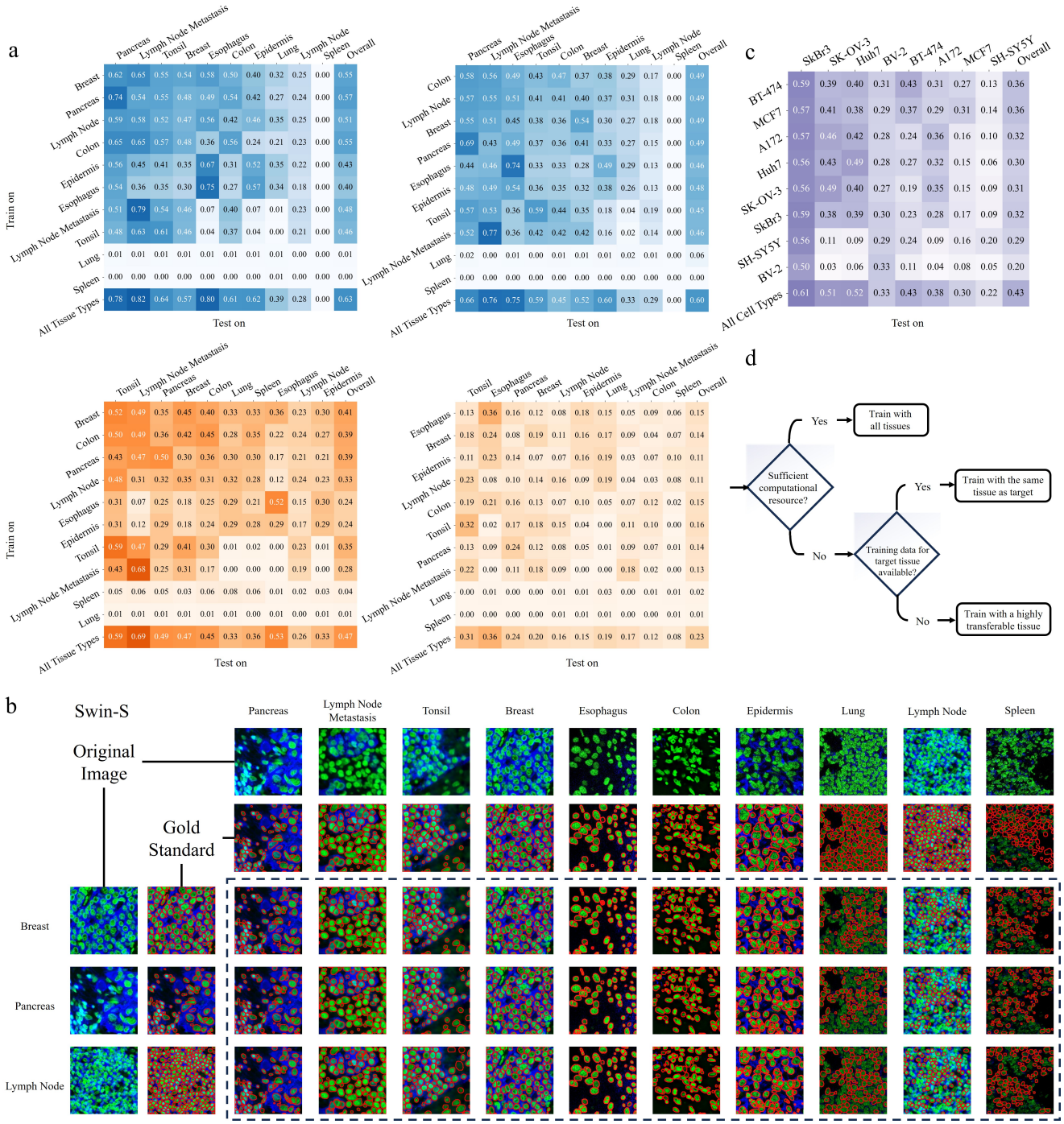

Figure 4. **a**, Evaluation results of Swin-S in TissueNet (First row from left to right: nuclei segmentation with dual-channel images and nuclei segmentation with nuclei images; Second row from left to right: whole cell segmentation with dual-channel images and whole cell segmentation with whole cell images) and **c**, Evaluation results from Swin-S in LIVECell (whole cell segmentation with whole cell images). They present the performance impact of cell segmentation when trained on a single tissue or cell type and tested on each tissue or cell type and all tissue or cell types denoted as *Overall*. The performance across all tissue or cell types is ranked in descending order for both rows and columns. Here, *All Tissue Types* or *All Cell Types* present the results trained on all tissue or cell types to facilitate an intuitive comparison against specialized training on one type. **b**, Visual results of Swin-S in nuclei segmentation with dual-channel images in TissueNet. The rows indicate the tissue types used for training, while the columns represent the tissue types used for testing. Only the first three rows are presented. **d**, The strategy for choosing training data in real practices.

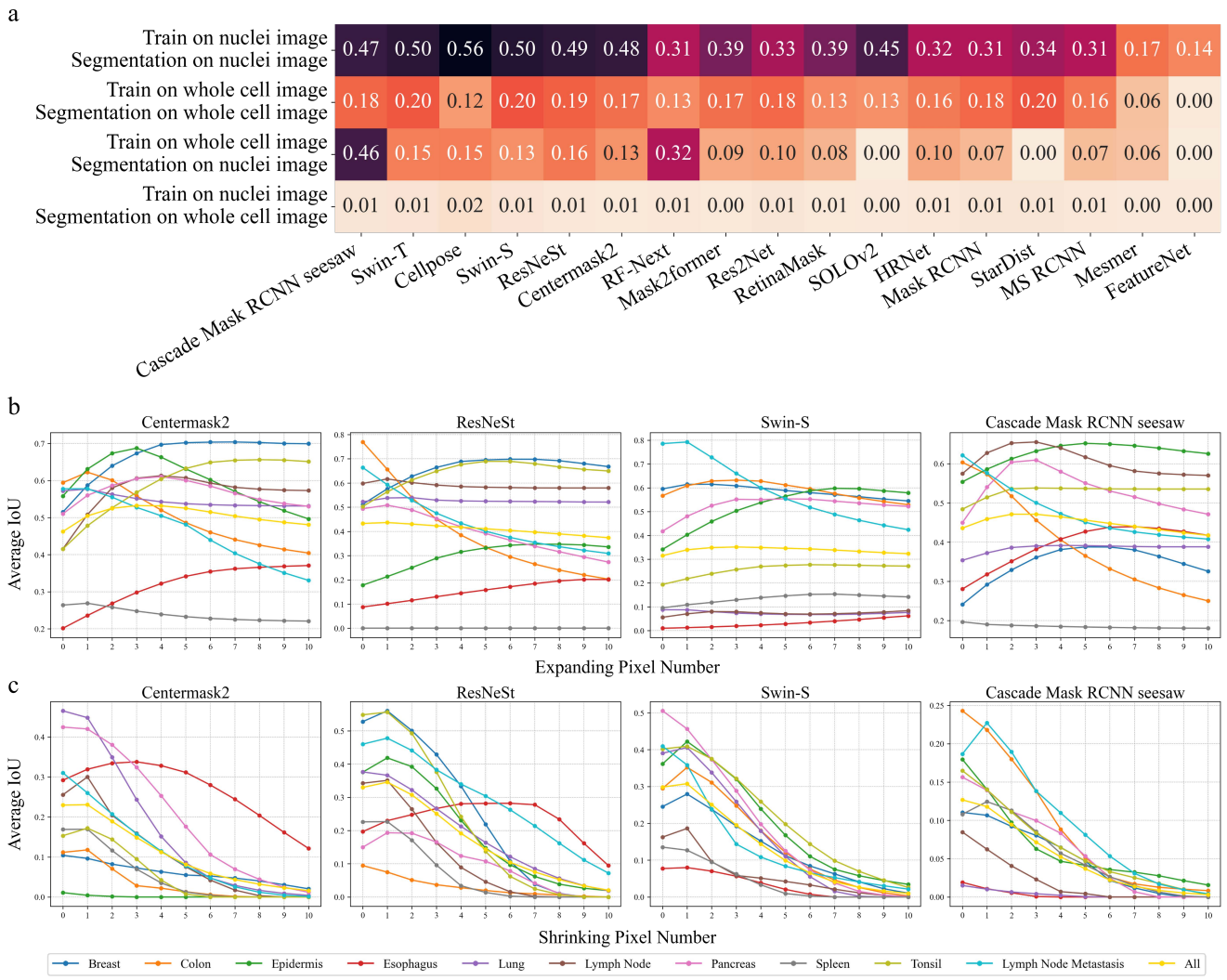

Figure 5. a, Generalization evaluation of segmentation models trained on one image modality and tested on a different image modality. **b**, Generalization evaluation of segmentation models when dilating cell nuclei segmentation to obtain whole cell segmentation. **c**, Generalization evaluation of segmentation models when eroding whole cell segmentation to obtain cell nuclei segmentation. The x-axis in **b** and **c** represents the number of pixels expanded or shrunk, ranging from 0 to 10, where 0 represents the original segmentation results, while the y-axis shows the average *IoU* scores between the expanded nuclei segmentation and the ground truth whole cell segmentation, and the shrunk whole cell segmentation and the ground truth nuclei segmentation, respectively.

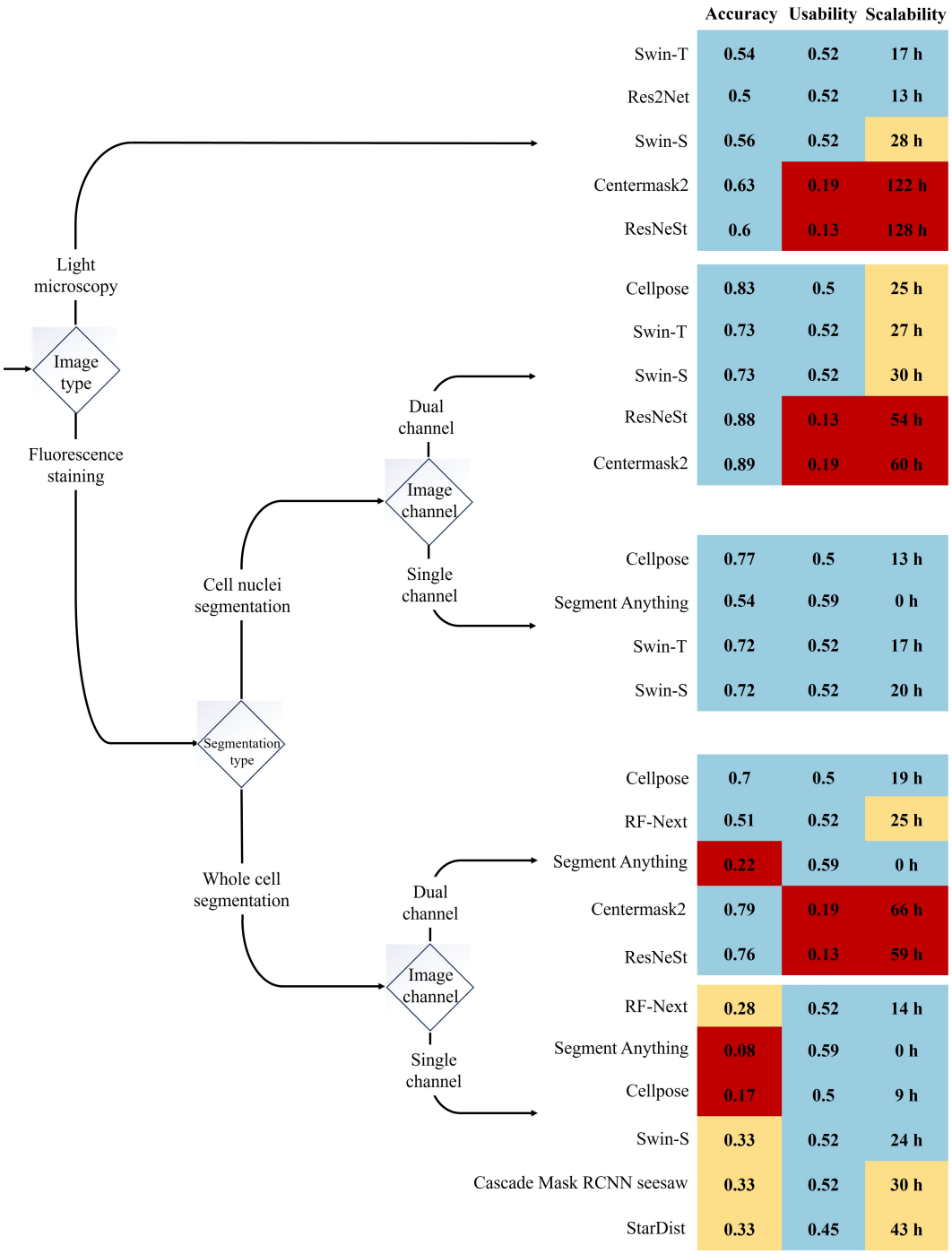

Figure 6. Practical guidelines for method users. The methods on the right are selected based on a union of the top three in overall performance and accuracy across respective experiments and then are ranked according to their overall performance considering accuracy, usability, and scalability. Further to the right shows accuracy (Blue : ≥ 0.5, Yellow : ≥ 0.25, Red : ≥ 0.0), usability (Blue : ≥ 0.5, Yellow : ≥ 0.25, Red : ≥ 0.0), scalability (Blue : ≤ 24, Yellow : ≤ 48, Red : > 48).

References

1. Bakal, C., Aach, J., Church, G. & Perrimon, N. Quantitative morphological signatures define local signaling networks regulating cell morphology. *science* **316**, 1753–1756 (2007).
2. Phillip, J. M., Han, K.-S., Chen, W.-C., Wirtz, D. & Wu, P.-H. A robust unsupervised machine-learning method to quantify the morphological heterogeneity of cells and nuclei. *Nat. protocols* **16**, 754–774 (2021).
3. Tegtmeyer, M. *et al.* High-dimensional phenotyping to define the genetic basis of cellular morphology. *Nat. Commun.* **15**, 347 (2024).
4. Zhao, S. *et al.* Single-cell morphological and topological atlas reveals the ecosystem diversity of human breast cancer. *Nat. Commun.* **14**, 6796 (2023).
5. Peng, H. *et al.* Morphological diversity of single neurons in molecularly defined cell types. *Nature* **598**, 174–181 (2021).
6. Akbari, P. *et al.* A genome-wide association study of blood cell morphology identifies cellular proteins implicated in disease aetiology. *Nat. communications* **14**, 5023 (2023).
7. Wang, Y. *et al.* Genesegnet: a deep learning framework for cell segmentation by integrating gene expression and imaging. *Genome Biol.* **24**, 235 (2023).
8. Edlund, C. *et al.* Livecell—a large-scale dataset for label-free live cell segmentation. *Nat. methods* **18**, 1038–1045 (2021).
9. Greenwald, N. F. *et al.* Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nat. Biotechnol.* **40**, 555–565 (2022).
10. Samacoits, A. *et al.* A computational framework to study sub-cellular rna localization. *Nat. Commun.* **9**, 4584 (2018).
11. Kandel, M. E. *et al.* Phase imaging with computational specificity (pics) for measuring dry mass changes in sub-cellular compartments. *Nat. communications* **11**, 6256 (2020).
12. Liberali, P., Snijder, B. & Pelkmans, L. Single-cell and multivariate approaches in genetic perturbation screens. *Nat. Rev. Genet.* **16**, 18–32 (2015).
13. Bunne, C. *et al.* Learning single-cell perturbation responses using neural optimal transport. *Nat. Methods* **20**, 1759–1768 (2023).
14. Lefebvre, A. E., Ma, D., Kessenbrock, K., Lawson, D. A. & Digman, M. A. Automated segmentation and tracking of mitochondria in live-cell time-lapse images. *Nat. Methods* **18**, 1091–1102 (2021).
15. Ulman, V. *et al.* An objective comparison of cell-tracking algorithms. *Nat. methods* **14**, 1141–1152 (2017).
16. Popescu, D.-M. *et al.* Decoding human fetal liver haematopoiesis. *Nature* **574**, 365–371 (2019).
17. Palla, G., Fischer, D. S., Regev, A. & Theis, F. J. Spatial components of molecular tissue biology. *Nat. Biotechnol.* **40**, 308–318 (2022).
18. Roerdink, J. B. & Meijster, A. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta informaticae* **41**, 187–228 (2000).

19. Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. Cellpose: a generalist algorithm for cellular segmentation. *Nat. methods* **18**, 100–106 (2021).

20. Schmidt, U., Weigert, M., Broaddus, C. & Myers, G. Cell detection with star-convex polygons. In *Medical Image Computing and Computer Assisted Intervention - MICCAI 2018 - 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II*, 265–273 (2018).

21. Lab, V. V. Deepcell-retinamask (2020). URL https://github.com/vanvalenlab/deepcell-retinamask/tree/master. GitHub repository.

22. Greenwald, N. F. *et al.* Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nat. biotechnology* **40**, 555–565 (2022).

23. Van Valen, D. A. *et al.* Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS computational biology* **12**, e1005177 (2016).

24. Lee, Y. & Park, J. Centermask: Real-time anchor-free instance segmentation. In *CVPR* (2020).

25. He, K., Gkioxari, G., Dollar, P. & Girshick, R. Mask r-cnn. *2017 IEEE Int. Conf. on Comput. Vis. (ICCV)* (2017).

26. Zhang, H. *et al.* Resnest: Split-attention networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2736–2746 (2022).

27. Huang, Z., Huang, L., Gong, Y., Huang, C. & Wang, X. Mask scoring r-cnn. In *IEEE Conference on Computer Vision and Pattern Recognition* (2019).

28. Cheng, B., Misra, I., Schwing, A. G., Kirillov, A. & Girdhar, R. Masked-attention mask transformer for universal image segmentation. *arXiv* (2021).

29. Wang, J. *et al.* Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2021).

30. Liu, Z. *et al.* Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030* (2021).

31. Wang, X., Zhang, R., Kong, T., Li, L. & Shen, C. Solov2: Dynamic and fast instance segmentation. *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)* (2020).

32. Gao, S., Li, Z.-Y., Han, Q., Cheng, M.-M. & Wang, L. Rf-next: Efficient receptive field search for convolutional neural networks. *IEEE Transactions on Pattern Analysis Mach. Intell.* **45**, 2984–3002 (2022).

33. Sun, K., Xiao, B., Liu, D. & Wang, J. Deep high-resolution representation learning for human pose estimation. In *CVPR* (2019).

34. Gao, S.-H. *et al.* Res2net: A new multi-scale backbone architecture. *IEEE TPAMI* (2020). DOI 10.1109/TPAMI.2019.2938758.

35. Kirillov, A. *et al.* Segment anything. *arXiv preprint arXiv:2304.02643* (2023).

36. Deng, J. *et al.* Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255 (Ieee, 2009).

37. Maška, M. *et al.* The cell tracking challenge: 10 years of objective benchmarking. *Nat. Methods* 1–11 (2023).

38. Caicedo, J. C. *et al.* Nucleus segmentation across imaging experiments: the 2018 data science bowl. *Nat. Methods* **16**, 1247–1253 (2019).

39. Ma, J. *et al.* The multi-modality cell segmentation challenge: Towards universal solutions. *arXiv:2308.05864* (2023).

40. Livecell dataset. URL https://sartorius-research.github.io/LIVECell/.

41. Tissuenet dataset. URL https://datasets.deepcell.org/data.

42. Lin, T.-Y. *et al.* Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755 (Springer, 2014).

43. Hirling, D. *et al.* Segmentation metric misinterpretations in bioimage analysis. *Nat. Methods* 1–4 (2023).

44. Vaswani, A. *et al.* Attention is all you need. *Adv. neural information processing systems* **30** (2017).

45. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 234–241 (Springer, 2015).

46. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).

47. Lin, T.-Y. *et al.* Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125 (2017).

48. Edlund, C. *et al.* LIVECell-A large-scale dataset for label-free live cell segmentation. *Nat. Methods* **18**, 1038–1045 (2021).

49. Gonzalez, R. C. *Digital image processing* (Pearson education india, 2009).

50. Pizer, S. M. *et al.* Adaptive histogram equalization and its variations. *Comput. vision, graphics, image processing* **39**, 355–368 (1987).

51. Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988 (2017).

52. Tian, Z., Shen, C., Chen, H. & He, T. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9627–9636 (2019).

53. Ren, S., He, K., Girshick, R. & Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. neural information processing systems* **28** (2015).

54. Cheng, B., Schwing, A. & Kirillov, A. Per-pixel classification is not all you need for semantic segmentation. *Adv. Neural Inf. Process. Syst.* **34**, 17864–17875 (2021).

55. Wang, X., Kong, T., Shen, C., Jiang, Y. & Li, L. SOLO: Segmenting objects by locations. In *Proc. Eur. Conf. Computer Vision (ECCV)* (2020).

56. Bodla, N., Singh, B., Chellappa, R. & Davis, L. S. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, 5561–5569 (2017).

57. Farha, Y. A. & Gall, J. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3575–3584 (2019).

58. He, K. *et al.* Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009 (2022).

59. Dosovitskiy, A. *et al.* An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

60. Tancik, M. *et al.* Fourier features let networks learn high frequency functions in low dimensional domains. *Adv. Neural Inf. Process. Syst.* **33**, 7537–7547 (2020).