Scellseg: A style-aware deep learning tool for adaptive cell instance segmentation by contrastive fine-tuning

Dejin Xun, Deheng Chen, Yitian Zhou, Volker M. Lauschke, Rui Wang, Yi Wang

ruiwang@zju.edu.cn (R.W.)
zjuwangyi@zju.edu.cn (Y.W.)

Highlights
Scellseg utilizes a style-aware pipeline to be more sensitive to different styles
Scellseg provides 3 benchmarked evaluation datasets with diverse biological levels
Scellseg achieves state-of-the-art transferability on three evaluation datasets
Scellseg explores the effects of pre-trained dataset and shot data
iScience

Article

Scellseg: A style-aware deep learning tool for adaptive cell instance segmentation by contrastive fine-tuning

Dejin Xun,1 Deheng Chen,2 Yitian Zhou,3 Volker M. Lauschke,3,4,5 Rui Wang,2,* and Yi Wang1,6,7,8,*

SUMMARY

Deep learning-based cell segmentation is increasingly utilized in cell biology due to the massive accumulation of large-scale datasets and excellent progress in model architecture and instance representation. However, the development of specialist algorithms has long been hampered by a paucity of annotated training data, whereas the performance of generalist algorithms is limited without experiment-specific calibration. Here, we present Scellseg, an adaptive pipeline that utilizes a style-aware pre-trained model coupled to a contrastive fine-tuning strategy that also learns from unlabeled data. Scellseg achieves state-of-the-art transferability in average precision and Aggregated Jaccard Index on disparate datasets containing microscopy images at three biological levels, from organelle, cell to organism. Interestingly, when fine-tuning Scellseg, we show that performance plateaued after approximately eight images, implying that a specialist model can be obtained with few manual efforts. For convenient dissemination, we develop a graphical user interface that allows biologists to easily specialize their self-adaptive segmentation model.

INTRODUCTION

Image-based single-cell profiling is widely used in biological, pharmaceutical, and medical applications, including in quantitative cytometry,1 spatial transcriptomics,2 high-content drug screening,3 and cancer metastasis analysis.4 However, due to a lack of robust and facile algorithms for single-cell analysis, average profiling remains the most commonly used method which may cause loss of information and mislead interpretation of feature associations.5 In recent years, deep learning has revolutionized the field of computer vision6 and catalyzed the advancement of single-cell segmentation methods.

Differences across cell types, microscopy instruments, treatment methods, imaging modalities, and staining protocols can generate microscopy images with considerable diversity. As a consequence, cell segmentation algorithms were mostly developed for specific datasets,7–9 and these methods performed poorly when applied to other styles of cell images. To overcome this limitation, generalist algorithms have been developed. In 2018, a data science bowl challenge tried to segment nuclei from a large number of different styles of microscopy images using 841 diverse images containing 37,333 nuclei.10 Inspired by this competition, Stringer et al. annotated 608 images containing more than 70,000 segmented objects and developed a generalist algorithm named Cellpose, which exhibited excellent performance in segmenting cell bodies from many image styles.11 Although deep learning-based generalist algorithms have achieved great success, the generalization ability of these algorithms is constrained by their training set. Generalist segmentation algorithms still lack the capability to be self-adaptive for all kinds of microscopy images.

Fine-tuning of pre-trained models has been successfully used in computer vision12–15 and natural language processing16,17 due to its lower input requirements and more rapid convergence to achieve better performance. For cell instance segmentation, only a few preliminary attempts were reported, such as fine-tuning of a nuclear segmentation model to satisfy different needs from distinct laboratories,18 or transferring a pre-trained model of in vitro images to in situ tissue images.19 However, these studies typically used nuclei images for pre-training and tested model transferability on different nuclei images, and specialized evaluation datasets for various microscopy instances such as C. elegans,20 are far from well developed and adequately studied. In addition, they did not notice which representation method would
allow the model to learn better information from instances, hence the model may not capture enough prior knowledge for fine-tuning on different kinds of microscopy images. Recently, Edlund et al. showed that a segmentation model pre-trained on phase-contrast cell images can achieve great performance on fluorescence cell images using transfer learning, indicating that it is possible that a well pre-trained model can transfer to different styles of microscopy images.21

In this work, we established an adaptive pipeline named Scellseg for cell segmentation algorithms, which includes a style-aware cell segmentation architecture based on attention mechanisms and hierarchical information to improve the extraction and utilization of style features. We further incorporate a contrastive learning strategy to leverage information from unlabeled and pre-trained data. To evaluate the generalizability of the pipeline, we benchmarked our model on three fundamentally different styles of data, including C. elegans, label-free phase-contrast cell images, and subcellular organelles. Scellseg achieved state-of-the-art transferability compared with previous tools. Furthermore, it is our first effort to estimate the minimal extent of data required for a satisfying fine-tuning model and to demonstrate how instance representation and pre-trained datasets can influence model transferability. To facilitate the uptake of this pipeline, we developed a graphical user interface (GUI) that can conduct annotation, fine-tuning, and inference, thus making the model accessible for a wide range of users without coding experience. The tool can be found at https://github.com/cellimnet/scellseg-publish.

**RESULTS**

**Design of Scellseg with pre-trained architecture and contrastive fine-tuning**

First, we established a pre-trained and fine-tuning pipeline for the cell segmentation model. For initial training, we utilized a dataset containing various cell types to build a generalist model. Generally, segmentation of untrained images by this model will exhibit limited performance, such as fail of detecting instances or boundary of segmentation instance. To improve the specificity of the model and avoid time-consuming retraining, several images from novel data pool can be annotated for fine-tuning the established model using a few new labeled data (shot data) (Figure 1A). This workflow generated a style-aware structure to better extract and comprehend style-related information and developed a new fine-tuning strategy based on contrastive learning to better make use of diverse data features, including unlabeled data (query data) and pre-trained data. The resulting model, which we named Scellseg, contains two branches, a mask branch to compute the segmentation map of input and a contrast branch to explore the information between three types of data (Figure 1B).

The mask branch was based on the Cellpose model which is a member of the U-Net22 family that consists of a downsampling pass that extracts features from input data, an upsampling pass that organizes different features to fit for the final task, and a concatenation operation that relays the information extracted from the downsampling process to the upsampling pass. For convenient adjustment, we redivided this structure. The last Conv Unit was split from the upsampling pass and named Tasker, and the left was named as Extractor, containing the downsampling, upsampling, and concatenation parts. To improve the sensitivity of the model for different styles, we added attention gates (AGs) when passing the features extracted from downsampling to the upsampling pass. These AGs give the feature map weights to highlight salient features useful for a specific task and suppress feature activation in irrelevant regions. We used dense units to consider the information from early upsampling layers, aiming to delineate accurate object boundaries. To consider different-level style information, we also fed corresponding hierarchical style embeddings into different-level dense units (Figure S1).

Unlike conventional fine-tuning strategies that only use labeled data, to augment data utilization, we developed a contrastive fine-tuning (CFT) method to employ information from either labeled data or unlabeled and pre-trained data based on contrastive learning. Seven common cellular styles of images were chosen from pre-trained data to form contrast data (Figure S2). In the contrastive fine-tuning process, the contrast branch is used to compute the respective style embeddings of these three data and then a contrast loss function was designed to minimize the difference between the embeddings of shot and query data from the same experiment while maximizing the difference between the embeddings of shot and contrast data (Figure 1B). This contrast loss was added into the segmentation loss function, and then the total loss optimized the model via backpropagation.

**Scellseg with contrastive fine-tuning achieves state-of-the-art transferability on three evaluation datasets**

To compare the transferability of Scellseg with other algorithms, we adopted three different datasets named BBBC010_elegans,20 LIVECell_bv2,21 and mito (in-house prepared dataset...
containing mitochondrial images), representing three biological levels of microscopy images from organism, cell to organelle (Figure 2A). In total, the datasets contained 230 images, containing 91,024 segmentation objects. We visualized the distribution of areas and numbers for cells per image (Figure S3). The average areas for the three datasets vary greatly, approximately 1000, 150, and 100,000, and the numbers of instances in each image range from 2 to 2,815. We used t-distributed stochastic neighbor embedding (t-SNE) to visualize the style embeddings (see definition in 11) of these evaluation datasets together with pre-trained datasets and noted that the style of data in each dataset was determinant in the major cluster (Figure 2B).

To compare the influence of different instance representations, we benchmarked Scellseg against five other models, U-Net2, U-Net3, StarDist,24 HoVer,25 and Cellpose (see STAR methods for more details of these models). These five models were set with identical network structure and pre-trained with the same dataset and training strategy. We used the training data of each dataset to fine-tune the model at ten different random states. Most models achieved great improvements after fine-tuning. For the
BBBC010_elegans dataset, models except StarDist yielded at least 35% higher average precision. For the U-Net3 model, the fine-tuning strategy even yielded a dramatic increase of 62.1% in average precision. The different models differed drastically in performance, and the representation of the Cellpose (used in the Cellpose and Scellseg model) outperformed the other methods. Scellseg with contrastive fine-tuning achieved the best performance on all three datasets, especially on the BBBC010_elegans dataset. At Figure 2. Transferability of Scellseg with contrastive fine-tuning on three evaluation datasets

(A) Example images of three datasets.
(B) Visualization of style embeddings of three datasets and the pre-trained dataset using t-SNE.
(C–E) Performance of different models on the BBBC010_elegans (C), LIVECell_bv2 (D), and mito (E) datasets. Different colors correspond to different models, the dotted lines denote the performance of applying models directly and the solid lines denote the performance after fine-tuning. For Scellseg, we use contrastive fine-tuning (CFT) and for others, we use the classic fine-tuning strategy (FT). We did not plot the line corresponding to AP@0.5 less than 0.01. Each pre-trained and fine-tuning pipeline was conducted 10 times at various random states, and error bars represent the mean ± SD. * indicates p value < 0.05, determined by two-way ANOVA followed by Sidak’s multiple comparisons test for Scellseg with CFT and Cellpose with FT.

BBBC010_elegans dataset, models except StarDist yielded at least 35% higher average precision. For the U-Net3 model, the fine-tuning strategy even yielded a dramatic increase of 62.1% in average precision. The different models differed drastically in performance, and the representation of the Cellpose (used in the Cellpose and Scellseg model) outperformed the other methods. Scellseg with contrastive fine-tuning achieved the best performance on all three datasets, especially on the BBBC010_elegans dataset.
the universally used intersection over union (IoU) threshold of 0.5, our Scellseg and Cellpose both achieved high average precision when segmenting *C. elegans* (0.882 for Scellseg; 0.868 for Cellpose), microglial cell BV-2 (0.783 for Scellseg; 0.784 for Cellpose), and mitochondria in cardiomyocytes (0.927 for Scellseg; 0.922 for Cellpose). In contrast, Scellseg performed considerably better at higher thresholds, such as 0.75 on all three datasets ([0.670, 0.493, and 0.634] for Scellseg compared to [0.587, 0.475, and 0.571] for Cellpose; Figures 2C–2E).

We compared the segmentation results of Scellseg with or without the contrastive fine-tuning strategy. As shown in Figure 3, it is clear that the fine-tuned model exhibited considerably better instance detection capability. Importantly, the fine-tuning strategy improved the ability to distinguish adjacent cells and allowed the segmentation of scattered mitochondria around the nuclei in the mito dataset. However, although our contrastive fine-tuning strategy preserves some of the partial generalization ability, all re-trained models suffered a sharp decline compared with the initial generalization ability (Figure S4).

**Comparison of generalization ability and transferability based on pre-trained dataset experiments**

To explore how the pre-trained dataset can influence model transferability, we used different subsets of the Cellpose training set. The initial subset (Sneuro) contains only one style of images from the Cell Image Library, then additional styles of images were sequentially added, such as fluorescent cells (Sfluor), nonfluorescent and membrane-labeled cells (Scell), other microscopy data (Smicro), and nonmicroscopy images (“Sgeneral”, corresponding to the full Cellpose train set; Figure 4A). In comparison, we randomly chose the same number of images from the whole dataset as above subsets and marked them as S89, S288, S381, S455, and S540 (S540 corresponding to the full Cellpose train set too).
We pre-trained Scellseg with Sneuro, Sfluor, Scell, Smicro, and S89, S288, S381, and S455 and first tested the 
generalization ability of each model by applying it directly without any adaptation on three evaluation data-
sets. For C. elegans, the model trained with Sneuro, Sfluor, and Scell does not result in successful recog-
nition until the pre-trained dataset contains microscopy instances with structures beyond cells. For small 
and round BV-2 cells, the generalization ability increased with the richness of the dataset from Sneuro to 
Smicro, but there was still a large gap compared to more diverse datasets (S89, S288, and S381). For the 
segmentation of mitochondria, surprisingly the model trained with Sfluor outperformed all others 
(Figures 4Ba and 4D).

Next, we tested the transferability of each model (Figures 4C and 4E). As expected, the transferability of 
Scellseg can increase as more and more different styles of microscopy images are added. On the 
BBBC010_elegans dataset, a model pre-trained on Sneuro achieves only very poor transfer performance 
(0.013 mAP), and performance increases substantially only after the addition of different styles of fluores-
cent images (0.436 mAP). As more cell-like images are added, performance can increase further to 0.554.
To explore the extent of annotated data required for fine-tuning, we performed a shot data scale experiment on these evaluation datasets. We set 10 scale levels, and for each number of training set images when fine-tuning, we randomly sampled 10 times from the training pool to fine-tune the model, followed by testing of transferability. For this evaluation, we focused on Scellseg with CFT and Cellpose with classical fine-tuning because these models clearly outperformed the other three algorithms. For all datasets, we observed that initial performance was relatively low with large variance (Figure 5A). As the number of training images increase, the performance improves drastically and the variance decreases while Scellseg significantly outperformed Cellpose. For BBBC010_elegans, LIVECell_bv2, and the mito dataset, Scellseg with CFT obtains [2.0%, 4.8%, and 18.5%] final improvement respectively compared to [4%, 6%, and 12.2%] for Cellpose with classical fine-tuning. We conducted curve fitting using the hyperbola function for each method per dataset for further inspection of the transferability across different training numbers. The results show that, when increasing the training number, different methods converged to different values and the mAP converged differently across datasets. For the mito dataset, the mAP persistently increased while for BBBC010_elegans, the rate of convergence was relatively fast, and regardless of the dataset, the performance plateaued at eight images. Similar results were obtained using the mean Aggregated Jaccard Index27 as a means to evaluate transfer performance (Figure S5). Therefore, it is suggested that approximately eight images are required to achieve satisfactory transfer learning based on the generalized model.

To verify the function of our contrastive fine-tuning strategy, we conducted ablation experiments. Importantly, our contrastive fine-tuning strategy outperformed Scellseg using the classic fine-tuning method at different training-number experiments on all three evaluation datasets (Figure 5B). Due to the similar performance of the model when “only” was trained on Smicro, we conducted the same shot data scale
experiments and ablation experiments as Scellseg pre-trained on the Sgeneral dataset and, excitingly, our style-aware pipeline worked and again outperformed Cellpose with the classic fine-tuning strategy (Figure S6).

Graphical user interface of Scellseg

To make Scellseg accessible to scientists without coding experience, we designed a GUI (Figure 6) with three functional parts, i) view and draw, ii) fine-tune, and iii) inference. For basic annotations, users can modify the mask of an instance directly at single-pixel resolution without deleting the whole mask. We also developed a cell list management system to help users edit the corresponding mask and provide annotations, thereby allowing the simultaneous provision of ground truth for segmentation and cell class prediction. Furthermore, users can easily save or load cell lists.

In the second module, users can fine-tune the pre-trained model with their own manually labeled data. The system allows users to choose a pre-trained model from Scellseg, Cellpose, and HoVer. Furthermore, each model can be combined with either the contrastive or classic fine-tuning strategy, presented above. It will not only give biologists and pathologists more flexibility and versatility for their image analysis tasks but also help algorithm engineers easily conduct experiments to study such pre-trained and fine-tuning pipelines. Finally, users can use the fine-tuned model to conduct inference either for one image or use batch inference. After annotation or inference, users can also acquire images of each single instance for further analysis.

DISCUSSION

Accurate cell instance segmentation is still a challenging task for many laboratories. Although generalist models have been developed, these typically require large annotated datasets, which is time- and labor-consuming in data collection, particularly when a large number of segmented objectives are supposed to be covered. To augment data utilization, we first established a pipeline for the fine-tuning of pre-trained cell segmentation algorithms. On this basis, we proposed a style-aware pipeline, yielding the best transferability on three different benchmarking datasets. Specifically, the work achieved four main innovations: First, we developed a style-aware pre-trained network architecture by introducing attention mechanisms and hierarchical information. Together with a contrastive fine-tuning strategy to leverage the information from both unlabeled and pre-trained data based on contrastive learning,28–32 Scellseg became more sensitive to different styles. Second, we present the first holistic comparison of the impact of different instance representations and pre-trained datasets on the generalizability and transferability of the model. Third, we
organized three benchmarking datasets. To the best of our knowledge, this is the first attempt to explore the transferability of generalist segmentation algorithms on microscopy images at different biological levels, which can be utilized in further segmentation algorithm development. Finally, we show that eight images are sufficient for Scellseg fine-tuning to achieve satisfactory performance based on a shot data scale experiment.

Importantly, after fine-tuning, AP for BBBC010_elegans and mito can reach up to approximately 0.9 at a threshold of 0.5, more than 37% and 36% improvements, respectively. As shown in Figure S7, five models give different methods during the process of mapping the raw mask to instance representation. Based on our benchmarking, topological maps generated by the Cellpose model constituted the best way to introduce rich instance information. In the future, the combination scheme design of these representations for the cell segmentation model deserves further study. Segmentation results could be further improved by our style-aware pipeline, which exhibited the best transferability on all three evaluation datasets, indicating that introducing such style relevant information can benefit the fine-tuning process. Notably, while generalization ability overall declined after fine-tuning to a specific task, our contrastive fine-tuning considerably improved generalizability. Emerging methods such as continual learning are also worth to investigating in this context.

In the pre-trained dataset scale experiments, we observed that the transferability of Scellseg-CFT increased with the richness of the pre-trained dataset, suggesting that our Scellseg-CFT pipeline can also benefit from large-scale and high-diversity datasets. Notably, the generalization ability of Scellseg pre-trained with SFluor containing different styles of fluorescent images outperformed all other models on the mito evaluation dataset, indicating that model pre-training on diverse but specialized data may yield greater performance than both low-diversity specialized datasets (such as Sneuro) or high-diversity generalized datasets (such as Scell that also contains nonfluorescent images). We also noted that the generalization ability increased with the addition of more different microscopy instances beyond cells to other noncell instances, such as C. elegans. For the shot data scale experiment, it is not surprising that performance increases with the number of training set images. However, what excited us is that we observed a large payoff when increasing the number of training images from 1 to 3, and performance plateaued after approximately eight images. These results are of high practical relevance as they indicate that few manual efforts are sufficient to yield a sufficiently fine-tuned model. Few-shot, one-shot, and zero-shot learning strategies can be studied to further reduce the number of annotated images needed. Notably, when the number of training set images is small, different training images can have very large impacts on the fine-tuning process, whereas we observed that as the number increases variance becomes substantially smaller. In the future, active learning on cell instance segmentation promises to refine shot data selection for fine-tuning.

Recently, Stringer et al. also preprint a work about fine-tuning of cell segmentation. Different from us, they want to clarify whether pre-trained models can achieve state-of-the-art performance faster with a human-in-the-loop approach primitively proposed for TissueNet, and the results are exciting. In our article, we clarify that a better instance representation, higher diversity, and larger scale of the pre-trained dataset are important for transferring to the target domain, and the performance can be further improved by our style-aware pipeline on some specialist datasets. We also explored the transferability of generalist segmentation algorithms on microscopic images at different biological levels. We open-source Scellseg with the hope that it can benefit the cell segmentation community together with similar works such as Cellpose 2.

By integrating attention mechanisms and hierarchical information for cell instance segmentation with a contrastive fine-tuning strategy, Scellseg features the highest transferability when benchmarked on three diverse imaging datasets against currently used segmentation methods. With an easy-to-use GUI, Scellseg can make advanced parallelized segmentation accessible to researchers without coding experience. Moreover, the Extractor and Tasker design can facilitate the adaptation to other computer vision tasks, such as segmentation and simultaneous class prediction, or conducting feature extraction for phenomics analysis. We anticipate that Scellseg will serve not only for cell segmentation but also for a wide range of other applications in cell biology and biomedicine.

Limitations of the study

However, in this work, we did not research the influence of the basic model backbone and all models were based on convolutional neural networks (CNNs). In recent years, self-attention architectures (such as
Transformer\textsuperscript{40} have shown great success and studies have attempted to apply them to computer vision.\textsuperscript{41} Such transformer architectures have better expressive ability but require more data for accurate training. Nevertheless, we believe that such approaches will eventually provide an important improvement in computer vision compared to CNN. Furthermore, we only benchmarked our algorithm on three representative datasets, and more datasets can be evaluated in future work.

**STAR METHODS**

Detailed methods are provided in the online version of this paper and include the following:

- **KEY RESOURCES TABLE**
- **RESOURCE AVAILABILITY**
  - Lead contact
  - Materials availability
  - Data and code availability
- **METHOD DETAILS**
  - Datasets
  - Models
  - Pre-training segmentation models
  - Fine-tuning segmentation models
  - Inference
  - Benchmarking
  - Shot data scale experiments
  - Hardware and computing environment used
- **QUANTIFICATION AND STATISTICAL ANALYSIS**

**SUPPLEMENTAL INFORMATION**

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2022.105506.

**ACKNOWLEDGMENTS**

The authors are grateful for the support from ZJU PII-Molecular Devices Joint Laboratory and support from “Medicine + X” interdisciplinary Center of Zhejiang University. Y.W. is supported by National Key R&D Program of China (2021YFC1712905), National Natural Science Foundation of China (No. 82173941), the Innovation Team and Talents Cultivation Program of National Administration of Traditional Chinese Medicine (No. ZYYCXTD-D-202002). R.W. is supported by National Natural Science Foundation of China (No. 61872319) and Natural Science Foundation of Zhejiang Provincial (No. LR18F020002).

**AUTHOR CONTRIBUTIONS**

Y.W. and R.W. proposed the concept and supervised project. R.W. provided guidance on pipeline development. Y.W., R.W., and D.J.X. designed the experiments. D.J.X. established the pipeline, organized datasets, conducted experiments, and performed data analysis. D.J.X. and D.H.C. developed the graphical user interface. D.J.X., V.M.L., Y.W., R.W., and Y.T.Z. wrote the manuscript.

**DECLARATION OF INTERESTS**

A patent covering all the main aspects of the use of ScellSeg as a tool for segmentation of cell-like objects has been filed by Zhejiang University (CNIPA application number: 2021115528528). The application is currently pending. D.J.X., R.W., and Y.W., as employees and student of Zhejiang University, are named as co-inventors on the patent application.

Received: June 16, 2022
Revised: October 27, 2022
Accepted: November 2, 2022
Published: December 22, 2022
REFERENCES

1. Cheng, S., Fu, S., Kim, Y.M., Song, W., Li, Y., Xue, Y., J., and Tian, L. (2021). Single-cell cytometry via multiplexed fluorescence prediction by label-free reflectance microscopy. Sci. Adv. 7, eabe0431. https://doi.org/10.1126/sciadv.abe0431.

2. Petukhov, V., Xu, R.J., Solidov, R.A., Cadun, P., Khodosevich, K., Mofllit, J.R., and Kharhouni, P.V. (2022). Cell segmentation in imaging-based spatial transcriptomics. Nat. Biotechnol. 40, 345–354. https://doi.org/10.1038/s41592-021-01044-w.

3. Moen, E., Bannon, D., Kudo, T., Graf, W., Covert, M., and Van Valen, D. (2019). Deep learning for cellular image analysis. Nat. Methods 16, 1233–1246. https://doi.org/10.1038/s41592-019-0403-1.

4. Pan, C., Schoppe, O., Parra-Damas, A., Cai, R., Todokoro, M.I., Gondi, G., von Neubeck, B., Buğurcu-Seidel, N., Seidel, S., Steineman, K., et al. (2019). Deep learning reveals cancer metastasis and therapeutic antibody targeting in the entire cell. Cell 179, 1661–1676.e19. https://doi.org/10.1016/j.cell.2019.11.013.

5. Rohban, M.H., Abbasi, H.S., Singh, S., and Carpenter, A.E. (2019). Capturing single-cell heterogeneity via data fusion improves image-based profiling. Nat. Commun. 10, 2082. https://doi.org/10.1038/s41467-019-10154-8.

6. Chai, J., Zeng, H., Li, A., and Ngai, E.W. (2018). Fine-tuning convolutional neural networks for fine art classification. Expert Syst. Appl. 114, 107–118. https://doi.org/10.1016/j.eswa.2018.07.026.

7. You, K., Kou, Z., Long, M., and Wang, J. (2020). Co-Tuning for transfer learning. Adv. Neural Inf. Process. Syst. 33, 17236–17246.

8. Krasowski, N.E., Beier, T., Knott, G.W., Kothe, M.I., Gondi, G., von Neubeck, B., Bügürçu-Seidel, N., Seidel, S., Steineman, K., et al. (2019). Deep learning reveals cancer metastasis and therapeutic antibody targeting in the entire cell. Cell 179, 1661–1676.e19. https://doi.org/10.1016/j.cell.2019.11.013.

9. Lee, C., Cho, K., and Kang, W. (2020). Mixout: effective regularization to facilitate large-scale pretrained language models. Preprint at arXiv. https://doi.org/10.48550/arXiv.1909.11299.

10. Zaki, O., Guda, P.R., Lee, K., Kim, J., Ozburn, L., Shachar, S., Gadkari, M., Sun, J., Fraser, I.D.C., Franco, L.M., et al. (2020). A deep learning pipeline for nucleus segmentation. Cytometry A, 97, 1248–1264. https://doi.org/10.1002/cyto.a.24257.

11. Jin, Y., Toberoff, A., and Azizi, E. (2021). Transfer learning framework for cell segmentation with incorporation of geometric features. Preprint at bioRxiv. https://doi.org/10.1101/2021.02.28.433289.

12. Wällby, C., Kamentsky, L., Liu, Z.H., Riklin-Raviv, T., Conery, A.L., O’Rourke, E.J., Sokolnicki, K.L., Visvikis, O., Ljosa, V., Irazoqui, J.E., et al. (2012). An image analysis toolbox for high-throughput C. elegans assays. Nat. Methods 9, 714–716. https://doi.org/10.1038/nmeth.1984.

13. Edlund, C., Jackson, T.R., Khalid, N., Bevan, N., Dale, T., Dengel, A., Ahmed, S., Tryggja, J., and Sigfusn, R. (2021). LiveCell—a large-scale dataset for label-free live cell segmentation. Nat. Methods 18, 1038–1045. https://doi.org/10.1038/s41592-021-01249-6.

14. Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (Springer), pp. 234–241.

15. Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. J. Mach. Learn. Res. 9, 2579–2605.

16. Schneider, U., Weigert, M., Broadau, C., and Myers, G. (2018). Cell detection with star-convex polygons. In Medical image computing and computer assisted intervention – MICCAI 2018. MICCAI 2018, 11071, pp. 265–273. Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-030-00934-2_30.

17. Graham, S., Yu, Q.D., Raza, S.E.A., Azam, A., Tsang, Y.W., Kwak, J.T., and Rajpoot, N. (2019). Hover-Net: simultaneous segmentation and classification of nuclei in multi-tissue histology images. Med. Image Anal 58, 101563. https://doi.org/10.1016/j.media.2019.101563.

18. Yu, W., Lee, H.K., Harirhan, S., Bu, W.Y., and Ahmed, S. (2020). Supervised contrastive learning for pre-trained language model fine-tuning. Preprint at arXiv. https://doi.org/10.48550/arXiv.2011.01403.

19. Yu, Y., Yuan, J., Jiang, H., Ren, W., Zhao, T., and Zhang, C. (2021). Fine-tuning pre-trained language model with weak supervision: a contrastive-regularized self-training approach. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Online: Association for Computational Linguistics), pp. 1063–1077.

20. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. (2020). What makes for good views for contrastive learning? In Advances in Neural Information Processing Systems 33, pp. 6827–6839.

21. Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. Proc. Mach. Learn. Res. 119, 1597–1607.

22. Zeng, G., Chen, Y., Cui, B., and Yu, S. (2019). Continual learning of context-dependent processing in neural networks. Nat. Mach. Intell. 1, 364–372. https://doi.org/10.1038/s42256-019-0080-x.

23. Das, D., and Lee, C.S.G. (2020). A two-stage approach to few-shot learning for image recognition. IEEE Trans. Image Process. 29, 3336–3350. https://doi.org/10.1109/TIP.2019.2959254.
35. Michaelis, C., Ustyuzhaninov, I., Bethge, M., and Ecker, A.S. (2019). One-shot instance segmentation. Preprint at arXiv. https://doi.org/10.48550/arXiv.1811.11507.

36. Xu, X., Tsang, I.W., and Liu, C. (2021). Complementary Attributes: a new clue to zero-shot learning. IEEE Trans. Cybern. 51, 1519–1530.

37. Zhou, Z., Shin, J.Y., Gurudu, S.R., Gotway, M.B., and Liang, J. (2021). Active, continual fine tuning of convolutional neural networks for reducing annotation efforts. Med. Image Anal. 71, 101997. https://doi.org/10.1016/j.media.2021.101997.

38. Stringer, C., and Pachitariu, M. (2022). Cellpose 2.0: how to train your own model. Preprint at bioRxiv. https://doi.org/10.1101/2022.04.01.486764.

39. Cuccarese, M.F., Earnshaw, B.A., Heiser, K., Fogelson, B., Davis, C.T., McLean, P.F., Gordon, H.B., Skelly, K.-R., Weathersby, F.L., Rodic, V., et al. (2020). Functional immune mapping with deep-learning enabled phenomics applied to immunomodulatory and COVID-19 drug discovery. Preprint at bioRxiv. https://doi.org/10.1101/2020.08.02.233064.

40. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems 30.

41. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin Transformer: hierarchical vision transformer using shifted windows. Preprint at arXiv. https://doi.org/10.48550/arXiv.2103.14030.

42. Ljosa, V., Sokolnicki, K.L., and Carpenter, A.E. (2012). Annotated high-throughput microscopy image sets for validation. Nat. Methods 9, 637. https://doi.org/10.1038/nmeth.2083.

43. Guerrero-Pena, F.A., Marrero Fernandez, P.D., Ing Ren, T., Yui, M., Rothenberg, E., and Cunha, A. (2018). Multiclass weighted loss for instance segmentation of cluttered cells. In 2018 25th IEEE International Conference on Image Processing (ICIP) (IEEE), pp. 2451–2455.

44. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). PyTorch: an imperative style, high-performance deep learning library. In Advances in neural information processing systems 32.
STAR METHODS

KEY RESOURCES TABLE

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---------------------|--------|------------|
| Deposited data      |        |            |
| Raw Cellpose dataset| http://www.cellpose.org | http://www.cellpose.org/dataset |
| Raw BBBC010_elegans dataset | Broad Bioimage Benchmark Collection | https://bbbc.broadinstitute.org/BBBC010 |
| Raw LIVECell_bv2 dataset | Dr. Richard Sjögren (Edlund et al., 2021) | https://sartorius-research.github.io/LIVECell |
| Evaluation datasets | This paper | https://scellseg-data.s3.cn-northwest-1.amazonaws.com.cn/evaluation_datasets.zip |

Software and algorithms

| Software and algorithms | Source | Identifier |
|-------------------------|--------|------------|
| Prism 8                 | GraphPad | https://www.graphpad.com/scientific-software/prism |
| PyCharm                 | JetBrains | https://www.jetbrains.com/pycharm |
| Python version 3.7      | Python Software Foundation | https://www.python.org |
| Cellpose                | Dr. Marius Pachitariu (Stringer et al., 2021) | https://github.com/MouseLand/cellpose |
| Raw StarDist            | Dr. Gene Myers (Schmidt et al., 2018) | https://github.com/stardist/stardist |
| Raw Hover-Net           | Dr. Nasir Rajpoot (Graham et al., 2019) | https://github.com/vqdang/hover_net |
| Scellseg                | This paper | https://github.com/cellimnet/scellseg-publish |

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Yi Wang (zjuwangyi@zju.edu.cn).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- The evaluation datasets have been deposited at Amazon and are publicly available as of the date of publication. The DOI is listed in the key resources table.
- All original code has been deposited at GitHub and is publicly available as of the date of publication. The DOI is listed in the key resources table.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Datasets

Pre-training datasets

We used the Cellpose dataset which contains a total of 608 images and over 70,000 segmented instances. 68 images were used for testing. Here, the whole training set (also named Sgeneral and SS40) was used to pre-train the models and the test set was used to evaluate the generalization ability in Figure S4. Furthermore, a subset of the training set containing a total of seven styles of images with five images per style was used as the contrast data (Figure S2).

Evaluation datasets

Three datasets were used to evaluate the transferability of different models, here called BBBC010_elegans, LIVECell_bv2 and mito. BBBC010_elegans was downloaded from the Broad Bioimage Benchmark Collection, containing 100 images of C. elegans in a screen to find novel anti-infectives. There are two
phenotypes in this dataset. For worms treated with ampicillin, they appear curved in shape and smooth in texture, while untreated worms appear rod-like in shape and slightly uneven in texture. Only the brightfield channel was used. We discarded images with heavily crossed instances because it is not the focus of our work, and the problem may be solved by some special postprocessing algorithm or introducing the z axis information when designing the ground truth. Finally, 49 images were reserved, 10 were used as the training set and 39 were used for testing.

The LIVECell_bv2 dataset consists of 536 phase-contrast images and over 330,000 segmented instances. These images were achieved using label-free phase-contrast imaging and cells in this dataset have small spherical morphology and are homogeneous across populations. Of the available images, 386 were used as the training set and 152 were used for testing.

We also generated a novel dataset called the mito dataset, which consisted of 49 fluorescent images of mitochondria from high content screening studies. The images were acquired by ImageXpress Micro Confocal (Molecular Devices). Each image contains two channels, a nuclear channel stained with Hoechst-33342 (Sigma) and a mitochondrial channel stained with tetramethylrhodamine methyl ester (TMRM, Sigma). All these images were manually annotated by a single human operator (D.J.X.), 10 images were used as the training set and 39 were reserved for testing. Because there was no clear boundary between individual cells.

All three datasets were organized in Cellpose format. The summary information can be seen in Table S1.

Models

How to represent instances is important for the cell segmentation model, and until now, to the best of our knowledge, there have been five main methods: U-Net2, U-Net3, StarDist, HoVer and Cellpose. In this article, U-Net2, U-Net3 and Cellpose refer to Cellpose project (https://github.com/MouseLand/cellpose), StarDist refers to https://github.com/stardist/stardist and HoVer refers to https://github.com/vqdang/hover_net. It is important to note that U-Net2 and U-Net3 carried out in Cellpose are different from their initial paper.

U-Net2 directly maps the annotated masks to 2 classes, where zero represents the background and one represents instances. This method usually performs poorly on touching cells because instance information was completely discarded. On the basis of U-Net2, U-Net3 adds distances to the cell boundary to distinguish between instances. In 2018, Schmidt et al. proposed localizing cell nuclei via star-convex polygons and achieved great success. In 2019, Simon et al. generated horizontal and vertical distance maps to provide rich instance information. In 2020, Stringer et al. generated topological maps through a process of simulated diffusion from masks. Here, we wanted to compare how expressive power different methods can provide, so we used the same network architecture as Cellpose. The Scellseg model adopted the representation of Cellpose because it had the best performance in the experiments. We plot a picture for each representation to help the readers clearly see the difference (Figure S7).

Two-channel 224 × 224 images were set as input for all models in this work. The primary channel contains instances to segment and the second optional channel can provide extra information such as the nuclei channel to support model learning. The hierarchical level of the Conv Units was set as [32, 64, 128, 256]. We computed the style embeddings through applying the average pooling on the feature map of the last Conv Unit and the dimensionality of each level style embeddings after being concatenated in the up-sampling pass is [256, 384, 448, 480].

Pre-training segmentation models

Pre-train different models with sgeneral

We trained six models (U-Net2, U-Net3, StarDist, HoVer, Cellpose, Scellseg) with Sgeneral, which contains 540 images, 64 of which were reserved for validation.

For U-Net2 and U-Net3, the loss is computed using the cross-entropy function, and a learning rate of 0.002 was selected to achieve good model convergence. For StarDist, the loss function refers to https://github.com/ASHISRAVINDRAN/stardist_pytorch/blob/master/distance_loss.py, and the learning rate was set to 0.0003. For HoVer and Scellseg, the loss function is the same as Cellpose, which was defined as:
\[ L_{\text{segmentation}} = \text{BCE}(y_{b,2}, \text{lbl}_{b,0}) + 0.5 \times \text{MSE}(y_{b,0.2}, 5 \times \text{lbl}_{b,1.3}). \]  
(Equation 1)

where BCE represents the binary cross-entropy loss, MSE represents the mean square error loss, \( y \) represents the prediction of the network, \( \text{lbl} \) represents the ground truth, subscripts correspond to the respective dimensions in \( y \) or \( \text{lbl} \), and \( b \) represents the batch size, which we set to 8.

All models were trained for 500 epochs (see also Cellpose) with stochastic gradient descent, the mean diameter was set to 30, and all other training hyperparameters were the same as Cellpose.

**Pre-trained dataset experiments**

We trained 8 other models across the different subsets of Cellpose mentioned above: Sneuro, Sfluor, Scell, Smicro, S89, S288, S381 and S455. For each subset, the last of every 8 images was reserved for validation.

The training logs of all models are shown in Figure S8.

**Fine-tuning segmentation models**

**Classic fine-tuning strategy (FT)**

When fine-tuning, the batch size was set to 8, the epoch was set to 100, the optimizer was Adam, the initial learning rate was set to 0.001 and every quarter of epochs it was reduced by 50%. Before being fed to the network, the images were normalized, resized, randomly rotated and reshaped with the ultimate shape of input as (8, 2, 224, 224).

**Contrastive fine-tuning strategy (CFT)**

The contrast loss function was defined as:

\[ L_{\text{contrast}} = \frac{\text{MSE}(\text{shot, query})}{\text{MSE}(\text{shot, contrast}) + 10^{-5}}. \]  
(Equation 2)

where MSE represents the mean square error and was used to compute the difference between embeddings, and \( 10^{-5} \) was added to prevent divisions by zero. This contrast loss was added into the segmentation loss function during contrastive fine-tuning, so the final loss function was defined as:

\[ L_{\text{total}} = L_{\text{segmentation}} + L_{\text{contrast}} \times \text{Sigmoid}(\alpha), \]  
(Equation 3)

where \( \alpha \) is a scalar to control the weight of contrast loss, also learnt during the fine-tuning process. A sigmoid function was used to ensure that the coefficient of contrast loss changed smoothly between zero and one. The initial \( \alpha \) in the contrast loss function was set to 0.2, and the initial learning rate of \( \alpha \) was set to 0.1, with reductions of 50% every quarter of epochs. Query and contrast images were normalized, resized, randomly cropped, randomly rotated and reshaped before being fed to the network. Other parameters were the same as for the classic fine-tuning strategy.

For both classic and contrastive fine-tuning strategies, we fine-tuned all layers because this method performed best compared with the downsampling part or the whole extractor (Figure S9). For each dataset, we computed the instance diameter using shot data without using the automated method provided by Cellpose, which was used in resizing the current diameter of instances to the mean diameter.

**Inference**

Then inferring, images are first normalized and resized before being fed into the network. Then, predictions output by the network needs to be processed to get final masks and different models adopt different approaches. The approaches for U-Net2, U-Net3 and Cellpose are provided by the Cellpose project at https://github.com/MouseLand/cellpose/blob/62d0e0460cc5e777bd0b557/bfcdf4c04e5627f9d/cellpose/utils.py#L325 and https://github.com/MouseLand/cellpose/blob/62d0e0460cc5e777bd0b557/bfcdf4c04e5627f9d/cellpose/dynamics.py#L489, the approaches for StarDist and HoVer refer to its corresponding project at (https://github.com/stardist/stardist/blob/a59f366f989a728813a8255323f20fa3d000e786/stardist/geometry/geom2d.py#L112 and https://github.com/vqdang/hover_net/blob/master/models/hovernet/post_proc.py), and the approach of Scellseg is the same as Cellpose.
Benchmarking

Metrics

We used average precision (AP) and the Aggregated Jaccard index (AJI) to evaluate segmentation performance (see 11 for detailed definitions). Except in Figures 2C–2E, we averaged the AP or AJI over IoU from 0.50 to 0.95 with a step size of 0.05 for convenient comparison and reserving the overall performance information simultaneously as detailed below:

\[
m\text{AP} = \frac{\text{AP}_{0.50} + \text{AP}_{0.55} + \cdots + \text{AP}_{0.90} + \text{AP}_{0.95}}{10},
\]  
\[
m\text{AJI} = \frac{\text{AJI}_{0.50} + \text{AJI}_{0.55} + \cdots + \text{AJI}_{0.90} + \text{AJI}_{0.95}}{10},
\]

(Equation 4)

(Equation 5)

Shot data scale experiments

We set a total of 10 scale levels; for BBBC010_elegans and mito we used [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] and for LIVECell_bv2, we used [1, 2, 4, 8, 16, 32, 64, 128, 256, 386]. For each experiment, we randomly sampled 10 times from the training set to fine-tune the pre-trained model. To eliminate issues due to different training data, the random state was kept identical across models. For example, we sampled the 9th, 5th, and 2nd images from the total of 10 images in the training set of a 3-shot experiment for the mito dataset, and then used the same images as training data for all five models.

Hardware and computing environment used

The code was written in Python programming language v.3.7.4. All experiments were conducted on an NVIDIA GeForce RTX 2080Ti. The deep learning framework used PyTorch44 v.1.7.1.

QUANTIFICATION AND STATISTICAL ANALYSIS

All figures were made using GraphPad PRISM 8.0 software (GraphPad Software, Inc., CA, USA). All graphs display mean values, and the error bars represent the standard deviation (SD). Statistical analyses were conducted with two-way repeated measures analysis of variance (ANOVA) followed by Sidak’s multiple comparisons test in Figures 2C–2E and two-way ANOVA in Figure 5A. A nonlinear regression curve fit was performed using a hyperbolic function in Figure 5A, given as:

\[
Y = \frac{B_{\text{max}}}{K_d + X},
\]

(Equation 6)

where Bmax and Kd are constants, for Scellseg-CFT, the (Bmax, Kd) of BBBC010_elegans, LIVECell_bv2 and mito are (0.5640, 0.04686), (0.4538, 0.1184), (0.5905, 0.5638) respectively, and for Cellpose-FT are (0.5155, 0.08557), (0.4473, 0.1496), (0.5458, 0.3555) respectively.