Microsnoop: a generalist tool for the unbiased representation of heterogeneous microscopy images

Authors

Dejin Xun[1], Rui Wang[2*], Xingcai Zhang[3*], Yi Wang[1,4,5*]

Affiliations

1. Pharmaceutical Informatics Institute, College of Pharmaceutical Sciences, Zhejiang University, Hangzhou, Zhejiang 310058, China

2. State Key Lab of CAD&CG, Zhejiang University, Hangzhou, Zhejiang 310058, China

3. John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA

4. Innovation Institute for Artificial Intelligence in Medicine of Zhejiang University, Hangzhou, Zhejiang 310018, China

5. State Key Laboratory of Component-based Chinese Medicine, Tianjin University of Traditional Chinese Medicine, Tianjin 300193, China

*Corresponding author. Email: zjuwangyi@zju.edu.cn (Y.W.); xingcai@mit.edu (X.C.Z.); ruiwang@zju.edu.cn (R.W.)

Abstract

Automated and accurate profiling of microscopy images from small-scale to high-throughput is becoming an essential procedure in basic and applied biological research. Here, we present Microsnoop, a novel deep learning-based representation tool trained on large-scale microscopy images using masked self-supervised learning, which eliminates the need for manual annotation. Microsnoop is able to unbiasedly profile a wide range of complex and heterogeneous images, including single-cell, fully-imaged and batch-experiment data. We evaluated the performance of Microsnoop using seven high-quality datasets, containing over 358,000 images and 1,270,000 single cells with varying resolutions and channels from cellular organelles to tissues. Our results demonstrate Microsnoop's robustness and state-of-the-art performance in all biological applications, outperforming previous generalist and even custom algorithms. Furthermore, we presented its potential contribution for multi-modal studies. Microsnoop is highly inclusive of GPU and CPU capabilities, and can be freely and easily deployed on local or cloud computing platforms.

MAIN TEXT

Introduction

Automated quantitative profiling of microscopy images has become increasingly prevalent in a wide range of biological studies, from small-scale to high-throughput research[1]. Visual phenotype or image representation, which entails profiling rich information from images, has proven to be beneficial in various areas of biology[2], such as identifying protein localization[3], classifying cell cycle stages[4], predicting mechanisms of action (MoA)[5], and high-content drug discovery[6]. Moreover, the growth of spatial omics has triggered new demand for the quantification of microscopy images. For instance, spatial proteomics technologies can image more than 50 disease-related proteins in a single tissue slice[7], while spatial transcriptomics allow for the simultaneous acquisition of image data and transcriptional profiles[8]. These advancements necessitate the development of a high-performance, generalist representation tool that is capable of handling heterogeneous microscopy images.

The profiling of microscopy images has been traditionally performed through extracting predefined morphological features, such as intensity, shape, texture, granularity, and colocalization[9-10]. However, these engineered features have limitations, including low computational efficiency, potential information loss, and sensitivity to image quality[11]. To address these deficiencies, recent advances in computer vision and deep learning have led to the development of learning-based feature extraction methods using representation learning, which involves pre-training a model on pretext tasks and then using part of the network as a feature extractor for downstream analysis. These methods can be divided into two categories: task-oriented custom methods and generalist methods. Task-oriented methods[4, 12-15] are pre-trained on data from the same source and developed specifically for biological research like cell cycle stage prediction. On the other hand, generalist methods require training data that are not focused on a particular biological problem. One of the most commonly used approaches is using models trained for ImageNet[16] (a natural image classification task), which continues to be used in recent multi-modal research[17]. However, whether the extent to which the feature extraction patterns are learned from natural images is sufficient for capturing the subtle phenotypes of microscopy images has not been fully validated by comparative research. In an effort to better match the feature domain to downstream microscopy image profiling tasks, the CytoImageNet[18] study was conducted, where image representation was learned based on a microscopy image classification task (890K images, 894 classes). Although this study demonstrated comparable performance to ImageNet, it still relied on the supervised learning approach that can be labor-intensive, prone to biases from semantic annotations, and potentially increase the difficulty to achieve higher representation performance.

The development of a high-performance, unbiased, generalist image representation tool can significantly increase the potential for advancement in the field of microscopy image analysis. Beyond facilitating accurate downstream analysis, such a tool would enable unsupervised analysis for identifying new phenotypes. It can facilitate the separation of feature extraction and downstream analysis process, allowing for downstream analysis conducted on computers with limited computing power. The representations of images that are much smaller than the original images can be easily stored and transferred, and private data can be shared securely through these representations without disclosing the original images. In addition, secondary analysis becomes possible, such as the creation of large image databases or joint analysis with other data representations. However, despite the above potential advantages, the complexity and diversity of microscopy images presents significant challenges in this tool development process.

Self-supervised representation learning offers the prospect of unbiased image representation by allowing the model to learn directly from the pixels without relying on pre-defined semantic annotations. This approach involves applying a transformation step to the original images and training the model to learn the mapping between the transformed and the original image. Transformation can take various forms, such as a direct copy[19], partial channel drop[20], or image masking[21], with masked visual representation learning being a particularly popular method in natural image studies[22-24]. Furthermore, recent advances in cell segmentation algorithms have indicated that networks trained on generalized data can possess remarkable generalization ability[25-27]. Despite these promising developments, there are several challenges to tackle to develop a universal tool for microscopy image profiling. These include handling images with varying resolutions and channel numbers (such as 1, 2, 3, 5 and 56)[3-4, 7, 26, 28]; requiring a single model to learn joint representation patterns for multiple image styles; processing various image types such as single-cell or fully-imaged images; or addressing technical variations in high-content experiments which may introduce batch effects in the feature space[29-30].

In this study, we present Microsnoop, a generalist tool for the unbiased representation of microscopy images based on masked self-supervised learning. Our pipeline was designed to handle heterogeneous images and includes a task distribution module to support users with different levels of computing power. To accommodate a broad range of image profiling needs, we have categorized images into three types and developed corresponding pipelines. We evaluated the performance of Microsnoop using seven evaluation datasets from diverse biological studies, comparing it to both generalist and custom algorithms. The results demonstrate the powerful feature extraction ability of Microsnoop and its potential for the analysis of multi-modal biological data. Our tool is freely available at https://github.com/cellimnet/microsnoop-publish.

Results

The design of a generalist representation tool.

It has been proved that a large and diverse dataset is beneficial for the training of generalist models. Herein, we collected and curated 10458 high-quality microscopy images published by the cell segmentation community[25-27, 31-33]. These images were taken using different technologies and have different resolutions and channel numbers, with channels ranging from cellular organelles to tissues. The four main types of images include fluorescence, phase-contrast, tissue and histopathology images (Fig. 1a(i) and Supplementary Table 1). To accommodate the variable number of image channels, the input to the neural network was set as one-channel images (related to one-channel feature concatenation strategy below). All images channels in the training set were split out and further selected to form a one-channel data pool (Methods). Before training, images in each batch were preprocessed in three steps: (1) Sample: randomly select one batch of images from the four types in turn to reduce the effects of unequal amounts of data; (2) Augment: randomly crop a 224*224 region (pad if smaller) from each image, then normalize, random rotate and scale the image, with the result serving as the network target; (3) Transform: randomly mask a portion of the target image patches, with the result serving as the network input. In terms of network architecture design, this study employed a CNN-based[34] (convolution neural network) architecture, despite the growing interest in Transformer-based architectures[35] for natural image analysis. This choice was motivated by the superior performance observed for the CNN architecture in our preliminary evaluations (Extended Data Fig. 1 and Methods). This performance disparity may be attributed to the difference in the amount of training data provided. Typically, the pre-training of a ViT architecture[36] requires a large corpus of data, with over 1 million or even 1 billion images used in the case of natural image studies[21]. However, our microscopy image dataset involved a relatively smaller set of training data, which may not have been sufficient to provide adequate training for the Transformer-based architecture.

We employed a masked self-supervised learning strategy to train the network, where a randomly selected percentage of image patches are masked and used as inputs. The network was then tasked with reconstructing the original, unmasked images. During training, masked images are encoded into high-level features through four consecutive downsampling steps, and the process of image reconstruction is accomplished through mirror-symmetric upsampling (Fig. 1a(ii)). The learning process is guided by minimizing the self-supervision loss function (Methods), which promotes the model to learn useful features that enable it to recover the masked parts of the images based on the information present in the remaining parts. This is a challenging task, which necessitates a comprehensive understanding that transcends simple low-level image statistics.

At test time, a generalist tool needs to face a range of image processing needs. To cater for this condition, we chose to categorize images based on the image profiling process itself, rather than solely on their biological applications that may be limited in scope. Our categorization comprises three types: single-cell images, fully-imaged images, and batch-experiment images.

(Fig. 1b(i)). The images to be processed are first managed by an in-built task distribution module (below), and then fed into the pre-trained encoder on a batch-by-batch basis for feature extraction. The output smallest convolutional maps are processed through global average pooling to produce initial 256-dimensional feature embeddings. Subsequently, feature aggregation is performed in accordance with different profiling tasks (details provided below). The final image representations can be used for various downstream analyses (Fig. 1b(ii)).

Diversified evaluation datasets.

In prior studies, attention was primarily focused on a limited number of specific datasets[5, 37-39]. In our work, to give a more comprehensive evaluation of our generalist tool, we collected and curated 7 evaluation datasets, encompassing commonly used datasets along with some novel additions, comprising over 358,000 images and 1,270,000 single cells (Methods and Extended Data Fig. 2). These images showcase a diverse array of characteristics, including various resolutions, image types, number of channels, and biological applications, such as protein localization estimation, cell cycle stage identification, and MoA prediction (Supplementary Table 2). In our study, four of the seven evaluation datasets focused on single-cell images. The performance of the model on fluorescent images, including bright-field channels, was assessed by COOS7 Test 1-4[39], CYCLoPs[3] and BBBC048[4]. For the assessment of the model's ability to handle more challenging histopathology images, we employed the CoNSeP[40] dataset. The LIVECell Test[26] and TissueNet Test[27] datasets were designed to evaluate a model's performance on fully-imaged image classification tasks, involving phase-contrast and tissue image representation, respectively. Lastly, the BBBC021[41] dataset was employed to evaluate the representation ability of the model for batch-experiment images.

Microsnoop accurately reconstructs the masked input images.

In the investigation of optimal mask ratio for learning features from microscopy images, we found that a 25% mask was optimal for this task. This was determined by testing 8 different mask ratios (5%, 15%, 25%, 35%, 45%, 55%, 65% and 75%) and comparing the results (Extended Data Fig. 3). To get a qualitative sense of the reconstruction task, we showed an example of each image type from the validation set (Fig. 2a). By inputting the 25% masked image into the pre-trained network, we were able to produce a reconstructed image that closely resembles the original, with only some detailed textures lost. This level of detail recovery is not easily achievable by humans. The reconstruction results on single-cell images from the evaluation datasets were even more impressive, with the reconstructed image being nearly indistinguishable from the original image (Fig. 2b and Extended Data Fig. 4). The improved performance on single-cell images in comparison to fully-imaged ones can be attributed to cellular heterogeneity, which results in diverse cell phenotypes. The abundance of reference information from single-cell images allows for the more successful recovery of a limited number of instances. These results demonstrate that the pre-trained Microsnoop network, has learned good representations of the microscopy images.

Microsnoop profile of single-cell images with one-channel feature concatenation.

Single-cell images can be produced directly by an imaging instrument such as imaging flow cytometry (IFC)[42], or obtained through cell segmentation processing on fully-imaged images. To accommodate the variable number of channels, we devised a one-channel feature concatenation strategy (Fig. 3a). Each channel of the multi-channel image is processed independently by Microsnoop, and the resulting embeddings are concatenated in an orderly manner. To prevent confusion during processing, a unique index is assigned to each image when multiple images are being processed. To address potential memory overflow issues when processing large batches of data, we established a task distribution module. This module efficiently manages image pathways and distributes images for processing, read into the CPU and transferred to the GPU as needed. The user is empowered to optimize performance by adjusting parameters according to the available memory capacity of both the CPU and GPU. Furthermore, our system features a scalable, distributed design, which is capable of supporting multiple GPUs, providing a solution for increasing data demands.

In our benchmark, we included three previous developed generalist methods in the comparisons: EfficientNetB0[43], Inception V3[44], CytoImageNet[18], and custom methods that are accessible (Methods). For the COOS7 Test 1-4, CYCLoPs and CoNSeP, we evaluated performance with the K-Nearest Neighbor (KNN) classification accuracy (match between prediction and ground truth using the KNN classifier, which has been utilized in prior study[18]). For the dataset BBBC048, we used fivefold cross-validation for dataset split and evaluated the performance with the multilayer perceptron (MLP) classification accuracy (match between prediction and ground truth using the MLP classifier, as employed in the original paper[4]). Our evaluations revealed the exceptional performance of Microsnoop, which consistently outperformed all other methods. In the majority of cases, Microsnoop achieved significant improvements of more than 6%, and up to 10% (Fig. 3b-f). Notably, for the 7-classification task of BBBC048, Microsnoop achieved an accuracy of 85.62% without using any data from the dataset, surpassing the custom supervised learning algorithm in the original paper by 5.02%.

Microsnoop profile of fully-imaged images with cell region cropping.

Fully-imaged images are a common format directly obtained from most microscopes. Cell segmentation is usually the first step of phenotype profiling due to the inherent heterogeneity of cells. Although various generalist segmentation algorithms[25-27] have been developed along with some fine-tuning strategies[45-46], they may still introduce unwanted segmentation errors. For instance, in a large drug screening experiment, cell body images can present a range of phenotypes, and a segmentation algorithm may perform well on some but poorly on others (Extended Data Fig. 5a), potentially leading to unpredictable impacts on downstream analysis. To mitigate these issues, we introduced a cell region cropping strategy, where the segmentation algorithm is applied only on the easiest channel, such as the nucleus channel, which presents more robust segmentation results (Extended Data Fig. 5b). Cell regions are computed and cropped based on the segmentation masks and rescale constant (Fig. 4a(i) and Methods). Then, Microsnoop extracts features from the cropped single-cell images as described above (Fig. 4a(ii)). Finally, the resulting single-cell level embeddings are aggregated by computing their mean to obtain the fully-imaged level representations (Fig. 4a(iii)).

We evaluated the representation ability of Microsnoop on two fully-imaged image phenotype classification tasks, and tested previously mentioned generalist algorithms for comparison. Both tasks were evaluated using the KNN classification accuracy. The results showed that Microsnoop again outperformed other methods, and even a 41.93% improvement was obtained on the LIVECell Test dataset (Fig. 4b-c). Furthermore, Microsnoop showed strong inclusiveness to various image styles, with an accuracy of 98.08% on the LIVECell Test dataset and 96.64% on TissueNet Test.

Microsnoop profile of batch-experiment images with sphering batch correction.

In high-content screening experiments, batch effects due to technical variability can affect downstream analysis[29-30, 37-38] (Fig. 5a). To address this issue, we employed a sphering batch correction method[47]. This assumes that the large variations observed in negative controls are associated with confounders, and any variation that is not observed in controls is associated with phenotypes. Sphering transformation aims to separate phenotypic variation from confounders. In our image representation pipeline for batch-experiment images, Microsnoop first extracts features from the fully-imaged images (as described above), and the resulting fully-imaged level representations are corrected via sphering transformation (Fig. 5b). Finally, the fully-imaged level representations are aggregated to treatment level representations by computing their mean (Fig. 5c).

We evaluated the representation ability of Microsnoop on the classic BBBC021 dataset, while including previously reported results of generalist and custom methods in the comparisons. We assessed the performance with the Not-Same-Compound (NSC) and Not-Same-Compound-or-Batch (NSCB) KNN classification accuracy. Microsnoop still achieved state-of-the-art performance without using any data from the dataset, even if compared with the methods exclusively studied on it (Fig. 5d-e).

Two other fully-imaged image profile modes and the robustness of cell region cropping mode.

In addition to the cell region cropping mode, we provided two alternative modes for processing fully-imaged datasets: rescaling and tile mode. In the rescaling mode, the shape of the fully-imaged images is directly rescaled to the input size (224*224) as inputs (Extended Data Fig. 6a-b). In the tile mode, the original image is cropped into multiple 224x224 tiles, and the fully-imaged level representations are aggregated by computing the mean over all tiles (Extended Data Fig. 6c). We evaluated the performance of these three processing modes, including different rescale constants for the cell region cropping mode, on both the fully-imaged and batch-experiment datasets (Extended Data Fig. 6d-g and Methods). The rescaling and tile modes outperformed the single-cell mode on LIVECell and TissueNet tests; however, both modes displayed a significant performance decline on the BBBC021 dataset. The reason for the underperformance of the rescaling mode could be attributed to the fact that it discards high-resolution phenotypic information during the rescaling process. On the other hand, the decline in performance observed with the tile mode may be due to the fact that it averages out important subtle phenotype variations present in certain regions of fully-imaged images. In contrast, the cell region cropping mode displayed robust performance across a range of parameters on all three datasets. Although the single-cell mode is more robust and efficient, it requires more time and memory compared to the other two modes. (Extended Data Fig. 6h-i).

Microsnoop improves the performance of the multi-modal structured embedding algorithm.

A recent study of the multi-modal structured embedding algorithm (MUSE[17]) has shown impressive results for the integrative spatial analysis of image and transcriptional data. The authors conducted a simulation experiment to assess the performance of MUSE when transcriptional data quality is degraded. Here, we focused on the impact of image feature quality, and the results of our simulation experiment showed that with the quality improvement of image representations, the performance of MUSE can also be significantly improved (Extended Data Fig. 7). Next, we tested Microsnoop on the real-world dataset seqFISH+[8] in comparison with the representation method used in the original paper. After acquiring the image representations, we use principal component analysis (PCA) performing feature dimensionality reduction to match the latent space dimensions of MUSE (Fig. 6a). We employed the silhouette coefficient[48] to evaluate the feature quality. Microsnoop demonstrated better image representation quality and greater improvement in the performance of MUSE (Fig. 6b).

Discussion

Advances in imaging technology, such as phase-contrast microscopy, imaging flow cytometry, automated high-throughput microscopy and microscopy combined with spatial omics techniques have created a massive demand to solve the complex challenge of microscopy image representation. In this study, we present Microsnoop, an innovative deep learning tool that effectively addresses this challenge. The accurate analysis of heterogeneous microscopy images, as a critical aspect of both fundamental and applied biological research, is highly valued by the microscopy image analysis community[49-50]. Our proposed solution offers promising advancements to this field. Microsnoop was trained on large-scale high-quality data using a masked self-supervised pretext task, allowing it to learn valuable and unbiased features for image representation. The one-channel feature concatenation strategy, efficient task distribution module, and rational classification mode of profiling needs make our tool flexible to meet various user needs. In addition, Microsnoop is capable of processing complex fully-imaged images through cell region cropping and mitigating batch effects in batch-experiment images through sphering transformation. For fully-imaged images, our results show that the single-cell analysis mode is more robust compared to other modes, reinstating the importance of considering cellular heterogeneity in biological research. Our benchmark results demonstrate robust and state-of-the-art performance on all evaluated datasets, eliminating the need to use of any evaluation data for fine-tuning. Furthermore, the enhanced representation of unimodal image data leads to significant improvements in the performance of multi-modal algorithms.

In our methodology experiments, we found that a mask ratio of 25% is optimal for microscopy images, which is significantly lower than the 75% that has been found optimal for natural images[21]. The difference is primarily due to the smaller size and erratic content of instances in microscopy images, which may result in lost information if too much reference information is masked. Compared with the CytoImageNet[18] study that utilized a supervised classification task as the pretext task, our masked self-supervised learning approach only requires raw images without any manual annotation and yields unbiased and more capable representations. Recently, a similar self-supervised representation learning study has also been reported as useful in learning the representations of protein subcellular location images through a pretext task that requires the network to directly reconstruct original images and images corresponding to similar proteins having similar representations[19]. In contrast, the uniqueness of our method is that ours do not require domain-specific knowledge and is developed for generalist image representation. Our benchmark study has shown that a single network is capable of handling heterogeneous microscopy images, which is in line with the conclusion reached in the sister domain of cell segmentation[25]. Furthermore, our pretext task was trained on the same network structure as Cellpose. This is reminiscent of the recent success of large pre-trained language models in the field of natural language processing[51-53]. With continued advancements in the understanding of computer vision and the further development of models for microscopy image representation and other image processing tasks, such as cell segmentation, it may be possible to merge these models into a single, unified model in the future.

While Microsnoop is a powerful tool, there are several areas for improvement. For example, further evaluation is needed to determine the efficacy of our approach of one-channel feature concatenation and feature aggregation in 3D and time-series imaging datasets in comparison to training a network to directly extract spatial or temporal information. To enhance the capabilities of Microsnoop, future work could include incorporating additional self-supervised pretext tasks for multi-task learning, optimizing the quality of the training dataset and refining the single-cell level feature aggregation methods. Moreover, the current training images are still limited in size compared to natural images, and a larger training data volume combined with the Transformer architecture can be studied to improve the performance. Last but not least, deploying our model on mobile devices to aid rapid detection could be a valuable application scenario[54].

Overall, we have developed an impressive, generalist tool for microscopy image representation. We anticipate its positive impact on the microscopy image analysis community, facilitating new phenotype discovery, data sharing, and the establishment of large image databases, among other benefits. Furthermore, we envision that Microsnoop can be effectively utilized in multi-modal studies such as combining molecular and image representation for MoA prediction[55-56] or exploring the relationship between gene expression and image representation for drug discovery[57].

References

1. Caicedo, J. C., Singh, S. & Carpenter, A. E. Applications in image-based profiling of perturbations. *Curr. Opin. Biotechnol.* **39**, 134-142 (2016).
2. Pratapa, A., Doron, M. & Caicedo, J. C. Image-based cell phenotyping with deep learning. *Curr. Opin. Chem. Biol.* **65**, 9-17 (2021).
3. Lu, A. X. et al. Integrating images from multiple microscopy screens reveals diverse patterns of change in the subcellular localization of proteins. e*Life* **7**, e31872 (2018).
4. Eulenberg, P. et al. Reconstructing cell cycle and disease progression using deep learning. *Nat. Commun.* **8**, 463 (2017).
5. Pawlowski, N., Caicedo, J. C., Singh, S., Carpenter, A. E. & Storkey, A. Automating Morphological Profiling with Generic Deep Convolutional Networks. Preprint at http://biorxiv.org/lookup/doi/10.1101/085118 (2016).
6. Cuccarese, M. F. et al. Functional immune mapping with deep-learning enabled phenomics applied to immunomodulatory and COVID-19 drug discovery. Preprint at http://biorxiv.org/lookup/doi/10.1101/2020.08.02.233064 (2020).
7. Schürch, C. M. et al. Coordinated Cellular Neighborhoods Orchestrate Antitumoral Immunity at the Colorectal Cancer Invasive Front. *Cell* **182**, 1341-1359 (2020).
8. Eng, C.-H. L. et al. Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+. *Nature* **568**, 235-239 (2019).
9. Carpenter, A. E. et al. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.* **7**, R100 (2006).
10. Pau, G., Fuchs, F., Sklyar, O., Boutros, M. & Huber, W. EBImage--an R package for image processing with applications to cellular phenotypes. *Bioinformatics* **26**, 979-981 (2010).
11. Singh, S., Bray, M.-A., Jones, T. R. & Carpenter, A. E. Pipeline for illumination correction of images for high□throughput microscopy. *J. Microsc.* **256**, 231-236 (2014).
12. Caicedo, J. C., McQuin, C., Goodman, A., Singh, S. & Carpenter, A. E. Weakly Supervised Learning of Single-Cell Feature Embeddings. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 9309-9318 (IEEE, 2018).
13. Lu, A. X., Kraus, O. Z., Cooper, S. & Moses, A. M. Learning unsupervised feature representations for single cell microscopy images with paired cell inpainting. *PLoS Comput. Biol.* **15**, e1007348 (2019).
14. Adnan, M., Kalra, S. & Tizhoosh, H. R. Representation Learning of Histopathology Images Using Graph Neural Networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 988-989 (IEEE, 2020).
15. Perakis, A. et al. Contrastive Learning of Single-Cell Phenotypic Representations for Treatment Classification. In *Machine Learning in Medical Imaging* (eds. Lian, C., Cao, X., Rekik, I., Xu, X. & Yan, P.) **12966**, 565–575 (Springer, 2021).
16. Russakovsky, O. et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **115**, 211-252 (2015).
17. Bao, F. et al. Integrative spatial analysis of cell morphologies and transcriptional states with MUSE. *Nat. Biotechnol.* **40**, 1200-1209 (2022).
18. Hua, S. B. Z., Lu, A. X. & Moses, A. M. CytoImageNet: A large-scale pretraining dataset for bioimage transfer learning. In *Proc. Advances in Neural Information Processing Systems* (Curran Associates, 2021).
19. Kobayashi, H., Cheveralls, K. C., Leonetti, M. D. & Royer, L. A. Self-supervised deep learning encodes high-resolution features of protein subcellular localization. *Nat. Methods* **19**, 995-1003 (2022).
20. Wong, D. R. et al. Trans-channel fluorescence learning improves high-content screening for Alzheimer's disease therapeutics. *Nat. Mach. Intell.* **4**, 583-595 (2022).
21. He, K. et al. Masked Autoencoders Are Scalable Vision Learners. In *Proc. IEEE Conference*

on Computer Vision and Pattern Recognition 16000-16009 (IEEE, 2022).

22. Liu, X., Zhou, J., Kong, T., Lin, X. & Ji, R. Exploring Target Representations for Masked Autoencoders. Preprint at https://arxiv.org/abs/2209.03917 (2022).
23. Li, Z. et al. MST: Masked Self-Supervised Transformer for Visual Representation. In *Proc. Advances in Neural Information Processing Systems 35* (Curran Associates, 2021).
24. Wei, C. et al. Masked Feature Prediction for Self-Supervised Visual Pre-Training. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 14668-14678 (IEEE, 2022).
25. Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods* **18**, 100-106 (2021).
26. Edlund, C. et al. LIVECell-A large-scale dataset for label-free live cell segmentation. *Nat. Methods* **18**, 1038-1045 (2021).
27. 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 (2021).
28. Bray, M.-A. et al. Cell Painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nat. Protoc.* **11**, 1757-1774 (2016).
29. Leek, J. T. et al. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat. Rev. Genet.* **11**, 733-739 (2010).
30. Lin, A. & Lu, A. X. Incorporating knowledge of plates in batch normalization improves generalization of deep learning for microscopy images. In *Proc. International Conference on Machine Learning* 74-93 (PMLR, 2022).
31. Kumar, N. et al. A Multi-Organ Nucleus Segmentation Challenge. *IEEE Trans. Med. Imaging.* **39**, 1380-1391 (2020).
32. Verma, R. et al. MoNuSAC2020: A Multi-Organ Nuclei Segmentation and Classification Challenge. *IEEE Trans. Med. Imaging.* **40**, 3413-3423 (2021).
33. Amgad, M. et al. NuCLS: A scalable crowdsourcing, deep learning approach and dataset for nucleus classification, localization and segmentation. *Gigascience* **11**, giac037 (2022).
34. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention* 234-241 (Springer, 2015).
35. Vaswani, A. et al. Attention is All you Need. In *Proc. Advances in Neural Information Processing Systems 30* (Curran Associates, 2017).
36. Dosovitskiy, A. et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations* (ICLR, 2021).
37. Ando, D. M., McLean, C. Y. & Berndl, M. Improving Phenotypic Measurements in High-Content Imaging Screens. Preprint at http://biorxiv.org/lookup/doi/10.1101/161422 (2017).
38. Bray, M.-A. et al. High-content cellular screen image analysis benchmark study. Preprint at https://www.biorxiv.org/content/10.1101/2022.05.15.491989v1.abstract (2022).
39. Lu, A. et al. The Cells Out of Sample (COOS) dataset and benchmarks for measuring out-of-sample generalization of image classifiers. In *Proc. Advances in Neural Information Processing Systems 32* (Curran Associates, 2019).
40. Graham, S. et al. Hover-Net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Med. Image Anal.* **58**, 101563 (2019).
41. Caie, P. D. et al. High-Content Phenotypic Profiling of Drug Response Signatures across Distinct Cancer Cells. *Mol. Cancer Ther.* **9**, 1913-1926 (2010).
42. Schraivogel, D. et al. High-speed fluorescence image-enabled cell sorting. *Science* **375**, 315-320 (2022).
43. Tan, M. & Le, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proc. International Conference on Machine Learning* 6105-6114 (PMLR, 2019).
44. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the Inception

Architecture for Computer Vision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 2818-2826 (IEEE, 2016).

45. Xun, D. et al. Scellseg: A style-aware deep learning tool for adaptive cell instance segmentation by contrastive fine-tuning. *iScience* **25**, 105506 (2022).
46. Pachitariu, M. & Stringer, C. Cellpose 2.0: how to train your own model. *Nat. Methods* **19**, 1634-1641 (2022).
47. Moshkov, N. et al. Learning representations for image-based profiling of perturbations. Preprint at http://biorxiv.org/lookup/doi/10.1101/2022.08.12.503783 (2022).
48. Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53-65 (1987).
49. Caicedo, J. C. et al. Data-analysis strategies for image-based cell profiling. *Nat. Methods* **14**, 849-863 (2017).
50. Chandrasekaran, S. N., Ceulemans, H., Boyd, J. D. & Carpenter, A. E. Image-based profiling for drug discovery: due for a machine-learning upgrade? *Nat. Rev. Drug. Discov.* **20**, 145-159 (2020).
51. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Preprint at https://arxiv.org/abs/1810.04805 (2018).
52. Brown, T. B. et al. Language Models are Few-Shot Learners. In *Proc. Advances in Neural Information Processing Systems 33* (Curran Associates, 2020).
53. Min, B. et al. Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey. Preprint at http://arxiv.org/abs/2111.01243 (2021).
54. Wang, B. et al. Smartphone-based platforms implementing microfluidic detection with image-based artificial intelligence. *Nat. Commun.* **14**, 1341 (2023).
55. Sanchez-Fernandez, A., Rumetshofer, E. & Hochreiter, S. CONTRASTIVE LEARNING OF IMAGE- AND STRUCTURE- BASED REPRESENTATIONS IN DRUG DISCOVERY. In *International Conference on Learning Representations* (ICLR, 2022).
56. Tian, G., Harrison, P. J., Sreenivasan, A. P., Puigvert, J. C. & Spjuth, O. Combining molecular and cell painting image data for mechanism of action prediction. Preprint at http://biorxiv.org/lookup/doi/10.1101/2022.10.04.510834 (2022).
57. Haghighi, M., Caicedo, J. C., Cimini, B. A., Carpenter, A. E. & Singh, S. High-dimensional gene expression and morphology profiles of cells across 28,000 genetic and chemical perturbations. *Nat. Methods* **19**, 1550-1557 (2022).
58. Ljosa, V., Sokolnicki, K. L. & Carpenter, A. E. Annotated high-throughput microscopy image sets for validation. *Nat. Methods* **9**, 637-637 (2012).
59. Hubert, L. & Arabie, P. Comparing partitions. *J. Classif.* **2**, 193-218 (1985).
60. Schneider, C. A., Rasband, W. S. & Eliceiri, K. W. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* **9**, 671-675 (2012).
61. Paszke, A. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proc. Advances in Neural Information Processing Systems 32* (Curran Associates, 2019).

Methods

Training set.

The training set consisted of four diverse image types from seven published datasets: Cellpose, LIVECell, TissueNet, and Histo, which includes MoNuSeg, MoNuSAC, and NuCLS. Firstly, all channels of the images were separated. For Cellpose and TissueNet, only the cell body channel was utilized, while the original RGB images of Histo were transformed into grayscale. The original training-validation dataset split was maintained for Cellpose, LIVECell, and TissueNet, while the images from the three Histo subsets were mixed and 20% were randomly reserved for validation purposes. Finally, the training set was organized into a one-channel image data pool. A comprehensive summary of the training set can be found in Supplementary Table 1.

Model architecture.

The network architecture was based on a refined version of the classic U-Net[34], as utilized in Cellpose. The standard convolutional blocks were replaced with residual blocks and style embeddings were incorporated into the concatenation stages. The downsampling scale was set as 32, 64, 128 and 256, and the upsampling scale was mirror symmetry. Both the input and output tensors were of shape batch_size*1*224*224 (in Pytorch tensor format, where batch_size is described below).

Masked self-supervised learning.

In the masked self-supervised learning approach, the network is tasked with reconstructing the original image from partial masked images. Our implementation involved dividing the target image (after normalization and augmentation) into 16*16 non-overlapping patches. Subsequently, a portion of these patches were randomly replaced with black patches of size 16*16, where every pixel was zero. Different from the original MAE built on a Transformer architecture, the transformed patches were restored to the image format to accommodate the input format of the CNN architecture.

Model training.

The self-supervision loss was set as the mean square error loss (MSE), which calculates the difference in both the masked and unmasked areas. The network was optimized by AdamW optimizer from the torch.optim Python package. In our implementation, we adopted a different definition of an epoch, in which one epoch corresponds to a complete iteration through all the sampled data, rather than through all the training data, as is commonly defined. During each epoch, we randomly sampled 12000 images from the four different types of training data in turn. The batch size was set as 16. The initial learning rate was set as 0.001, and we used a learning rate (LR) warmup trick: at the first 40 epochs, the LR was computed as:

$$\mathrm{LR} = 0.001 * \frac{\mathrm{epoch}}{40}$$

after 40 epochs, the LR was computed as:

$$\mathrm{LR} = 0.001 * 0.5 * [1 + \cos\left(\frac{\mathrm{epoch} - 40}{\mathrm{nepoch} - 40} * \pi\right)]$$

where nepoch represents the epoch size of the training process, here it was set as 1000.

One-channel feature concatenation strategy for multi-channel image representation.

In our implementation of Microsnoop for feature extraction, we assumed that the input data comprised multi-channel images with the same number of channels, represented as (c, h, w), where c denotes the number of channels, and h and w denote the height and width, respectively. In the event that images had different h and w, we padded them with zeros to obtain a consistent shape. The task distribution module is then used to read the images into CPU memory, where they are transformed into an array with shape (n, c, h, w), where n denotes the number of images read. This array is then reshaped into (n*c, 1, h, w), with each image assigned a unique index represented as a shape (n*c,) vector. For each batch of size b, the task distribution module transfers b images into the GPU memory, resulting in a tensor of shape (b, 1, h, w). After Microsnoop processes all n*c images, the CPU cache is cleared using the collect function from the gc Python package, and the next n images are read. The resulting embedding array had the shape of (N*c, 256), where N denotes the total number of processed images, and 256 is the pre-set dimensionality of the feature vector for a one-channel image in Microsnoop. These embeddings are then concatenated in channel to obtain a final feature embedding array of shape (N, 256*c).

Evaluation datasets.

We curated seven evaluation datasets, four of which were directly available from public sources and three (CoNSeP, LIVECell Test and TissueNet Test) were processed by us based on publicly acquired images. The summary of these datasets can be seen in Supplementary Table 2.

COOS7. This dataset contains 132,209 single-cell fluorescence images, including a training set and four test sets that vary in different factors. The training set consists of images from 4 independent plates, while Test 1 includes randomly held-out images from the same plates as the training set, Test 2 includes images from the same plates but different wells, Test3 comprises images produced months later, and Test 4 has images produced by other instruments. The images were downloaded through the link provided by Stanley Bryan Z. Hua[18]. Each image takes the shape of 2*64*64 and is a pixel crop centered around a unique mouse cell. One channel marks the protein targeting a specific component of the cell and the other marks the nucleus. There are 7 protein location classes in each set: Endoplasmic Reticulum, Inner Mitochondrial Membrane, Golgi, Peroxisomes, Early Endosome, Cytosol and Nuclear Envelope, and the evaluation task requires the model to accurately predict the protein location.

CYCLoPs. This dataset consists of 28,166 single-cell fluorescence images from the CYCLoPs database, and we downloaded the data through the link provided by Stanley Bryan Z. Hua[18]. Each image has a shape of 2*64*64 and is a pixel crop centered around a unique yeast cell. One channel marks the protein location and the other marks the cytosol. There are 17 protein location classes: ACTIN, BUDNECK, BUDTIP, CELLPERIPHERY, CYTOPLASM, ENDOSOME, ER, GOLGI, MITOCHONDRIA, NUCLEARPERIPHERY, NUCLEI, NUCLEOLUS, PEROXISOME, SPINDLE, SPINDLEPOLE, VACUOLARMEMBRANE and VACUOLE. The aim of the evaluation is to accurately predict the protein localization.

CoNSeP. This dataset has 41 H&E stained fully-imaged images with a shape of 3*1000*1000 pixels. 14 of these are test images and 27 are training images. The raw data were obtained from https://warwick.ac.uk/fac/sci/dcs/research/tia/data and then transformed into grayscale format. Each cell was cropped based on the provided segmentation mask, resulting in 8777 single-cell test images and 15554 single-cell training images with a shape of 1*112*112 pixels. In cases where the cells were smaller, padding was applied to obtain the desired size. The class information was extracted from the classification mask, with 4 classes: Other, Inflammatory, Epithelial, Spindle-shaped. The evaluation task requires the model to accurately predict the cell types.

BBBC048. This dataset contains 32,266 single-cell images from the Broad Bioimage Benchmark Collection[58]. These single-cell images of Jurkat cells were directly captured with the ImageStream imaging flow cytometer. Each image has a shape of 3*66*66 pixels, with a brightfield channel and two fluorescence channels. There are 7 cell phases: G1, S, G2, Prophase, Metaphase, Anaphase and Telophase. Another 5-phase case considers G1, S and G2 phase as a single class. The evaluation task requires the model to accurately predict the cell cycle stages.

LIVECell Test. This dataset comprises 1512 fully-imaged phase-contrast images provided by Christoffer Edlund[26], where each image has a shape of 1*520*704 pixels. There are 8 cell types: A172, BT474, BV2, Huh7, MCF7, SHSY5Y, SkBr3 and SKOV3. The evaluation task requires the model to accurately predict the cell types of full-imaged images.

TissueNet Test. This dataset comprises 1249 fully-imaged tissue images provided by Noah F. Greenwald□. Each image has a shape of 2*256*256 pixels, one channel marks the membrane or cytoplasm and the other marks the nucleus. We extracted the tissue type information from the metadata provided. There are 6 tissue types: Breast, Gi, Immune, Lung, Pancreas and Skin. The evaluation task requires the model to accurately predict the tissue types of full-imaged images.

BBBC021. This dataset includes 3848 fully-imaged fluorescence images, a subset from the Broad Bioimage Benchmark Collection[58]. The images are of MCF-7 breast cancer cells with a collection of 113 small molecules at different concentrations and a DMSO negative control. Each image has a shape of 3*1024*1280 pixels, and different channels respectively mark the DNA, F-actin and B-tubulin. There are 12 mechanisms: Actin disruptors, Aurora kinase inhibitors, Cholesterol-lowering, DNA damage, DNA replication, Eg5 inhibitors, Epithelial, Kinase inhibitors, Microtubule destabilizers, Microtubule stabilizers, Protein degradation and Protein synthesis. The evaluation task requires the model to accurately predict the MoA of different treatments.

Three modes for the profile of fully-imaged images.

Cell region cropping mode. We utilized the generalist tool Cellpose on the easiest channel (such as the nucleus channel) to perform cell segmentation. For each image, following the acquisition of the segmentation mask, we extract all the (x, y) pixel coordinates of each cell, and compute the region of each cell as follows:

$$w = x_{max} - x_{min}\ ;\ \mathrm{h} = y_{max} - y_{min}$$
$$x_c = x_{min} + 0.5 * \mathrm{w}\ ;\ \mathrm{y}_c = y_{min} + 0.5 * \mathrm{h}$$
$$\mathrm{Rs} = \min(\max(\mathrm{w}, \mathrm{h}) * \mathrm{Rc}, \mathrm{Sta} * 0.5)$$
$$\mathrm{bbox}_0 = \max\,(x_c - Rs, 0)\ ;\ \mathrm{bbox}_1 = \max\,(y_c - Rs, 0)$$
$$\mathrm{bbox}_2 = \min(x_c + Rs\,, \mathrm{W})\ ;\ \mathrm{bbox}_3 = \min(\,y_c + Rs\,, \mathrm{H})$$

where $x_{max}, x_{min}, y_{max}, y_{min}$ denote the max/min x/y, respectively, among all the pixels coordinates; x_c, y_c denote the coordinates of centroid; Rc denotes the rescale constant (it is set by user according to the average size of cell bodies); Sta denotes the side length of cropped image (here we set it as 224, the input size of Microsnoop); Rs denotes the crop size (it cannot be more than half of Sta); W, H denote the width and height of the fully-imaged image, respectively. $\mathrm{bbox}_o, \mathrm{bbox}_1, \mathrm{bbox}_2, \mathrm{bbox}_3$ denote the left, up, right, down of the cropped region in the original image, respectively, and they cannot go beyond the boundaries of the image. Finally, single-cell images are cropped on all channels and padded to (c, Sta, Sta) with zero pixels if smaller, where c denotes the number of channels. The fully-imaged level embedding of the image is obtained by computing the mean of all single-cell image embeddings.

Rescaling mode. In the case that the height of the image is not equal to its width, the initial step is to pad the image with zeros to create a square shape. The fully-imaged images are then rescaled to input size using the resize function from the cv2 Python package. The fully-imaged level embedding of the image is directly obtained through this process.

Tile mode. The fully-imaged images are cropped into tiles using the make_tiles function from the cellpose.transforms Python package. The parameter bsize was set as the input size, and the parameter tile_overlap was set as 0.1. The fully-imaged level embedding of the image is obtained by computing the mean of all tile embeddings.

Sphering transformation for the profile of batch-experiment images.

The detailed description can be found in ref. [47]. Here, we fitted the ZCA_corr transformer from https://github.com/jump-cellpainting/2021_Chandrasekaran_submitted/blob/main/benchmark/old_notebooks/utils.py on the embeddings of negative control, and then used the fitted transformer to correct the embeddings of each batch.

Benchmarking.

For BBBC021, we directly adopted the previously published state-of-the-art (SOTA) results from the curated resource at https://bbbc.broadinstitute.org/BBBC021. We also included the results of recently reported generalist methods. All results were formatted to two decimal places.

For other datasets, we compared with three generalist deep-learning methods: EfficientnetB0, Inception V3 and CytoImageNet. EfficientnetB0 was pretrained on the ImageNet and was included in the comparison in CytoImageNet. The famous project DeepProfiler[47] also used this network for the profiling of microscopy imaging data. Inception V3, which was also pre-trained on ImageNet, had been utilized in the MUSE project, a study of advanced multimodal algorithms. CytoImageNet, a recently published generalist microscopy image representation learning algorithm, was pre-trained using a self-constructed microscopy image classification dataset.

The results of EfficientnetB0 and CytoImageNet on COOS7 and CYCLoPs have been previously reported[18] and were directly adopted from the relevant publication. For BBBC048, we also included the custom algorithm results reported in the original paper. The remaining results presented in this paper were generated by the authors.

EfficientnetB0 and CytoImageNet were established using the EfficientnetB0 class from the tenforflow.keras.applications Python package, with different weights loaded (EfficientnetB0 used the ImageNet weights and CytoImageNet used the weights published by Stanley Bryan Z. Hua). Inception V3 was established using inception_v3 class from the torchvision.models Python package. We dropped the last classification layer and used the remaining network for feature extraction. Because these network architectures are presented in natural RGB image study, at test time, each one-channel image is copied three times to mimic RGB images (also used in ref. [18, 37]). The other steps, such as data preprocessing and feature aggregation, are identical to those used in the Microsnoop protocol.

For LIVECell and TissueNet Test, we directly used the provided segmentation masks (nucleus channel for the TissueNet) without applying the cell segmentation algorithm in the cell region cropping mode. For the COOS7, CYCLoPs and BBBC021 datasets, the number of nearest neighbors (k) in the KNN classifier was set to 11, 11, and 1, respectively, in accordance with the ref. [18]. For BBBC048, the MLP was conducted using the MLPClassifier class from the sklearn.neural_network Python package, and the parameter max_iter was set as 1000.

Joint use of Microsnoop and MUSE.

In the simulation experiment, we utilized the simulation_tool.multi_modal_simulator function from the MUSE project to generate the transcriptional and image representations along with the corresponding ground truth. We used the adjusted Rand index (ARI)[59] to assess the ability of discovering true subpopulations. For the analysis of seqFISH+ data, the microscopy images were provided by the authors of the seqFISH+ paper. Each cell region of the images was determined by the coordinates of the cell centroid provided. We used Microsnoop and Inception V3 to conduct feature extraction on the Nissl and DAPI stained images separately. The shape of each single-cell embedding output was 512 (256*2), then we used PCA to reduce the feature dimensionality to 500. The process of the transcript data was the same as MUSE. We used the silhouette coefficient to assess feature quality by the compactness of the clusters, which was conducted using the silhouette_score function from the sklearn.metrics Python package.

Graph plotting

All bar graphs were plotted using GraphPad PRISM 8.0 software (GraphPad Software, Inc., CA, USA). Fig. 1b(i) and Fig. 5a were created using resources from BioRender.com. The sources of images in Fig. 1 also included https://www.rxrx.ai/rxrx2, in addition to those listed in the supplementary Table 1 & 2. Some microscopy images in the figures have been processed using “Enhance Contrast…” from ImageJ[60] for better presentation.

Software and hardware

The programming was conducted using Python v.3.7. Training and all evaluations were performed on NVIDIA GeForce RTX 3090 GPUs. The deep learning framework of Microsnoop used PyTorch[61] v.1.10.

Data availability

The links to download the raw data of training set and evaluation datasets are provided in Supplementary Table 1-2. The new evaluation datasets generated by this study will be made available on figshare: https://figshare.com/articles/dataset/Microsnoop_a_generalist_tool_for_the_unbiased_representation_of_heterogeneous_microscopy_images/22197607 upon publication.

SeqFISH+ mouse cortex dataset: Transcript data were downloaded from https://github.com/CaiGroup/seqFISH-PLUS. Image data were provided by the authors of the seqFISH+ paper.

All data in this study are available from the corresponding author upon reasonable request.

Code availability

Source code for Microsnoop, including detailed tutorial, will be made available on GitHub (https://github.com/cellimnet/microsnoop-publish) upon publication. A configured Amazon Machine Image (AMI) will be made available upon publication for quickly and conveniently deploying Microsnoop for microscopy image analysis.

Acknowledgments

We thank L. Cai at Caltech for providing seqFISH+ image data. We thank W.K. Wang and L. Sun at Amazon Web Services China for their indispensable support in terms of computing resources and technology. We are grateful for the support from ZJU PII-Molecular Devices Joint Laboratory and support from "Medicine + X" interdisciplinary Center of Zhejiang University. This study was supported by National Key R&D Program of China (2021YFC1712905), National Natural Science Foundation of China (No. 82173941, No. 61872319), Key R&D Program of Zhejiang Province (No. 2023C01039). Y.W. was supported by the Innovation Team and Talents Cultivation Program of National Administration of Traditional Chinese Medicine (No. ZYYCXTD-D-202002).

Author contributions

Y.W., X.C.Z. and R.W. supervised the study, D.J.X. acquired data, established pipeline, conducted experiments and performed data analysis. D.J.X., Y.W., X.C.Z. and R.W. wrote the manuscript.

Competing interests

The authors declare no competing interests.

Figures and Tables

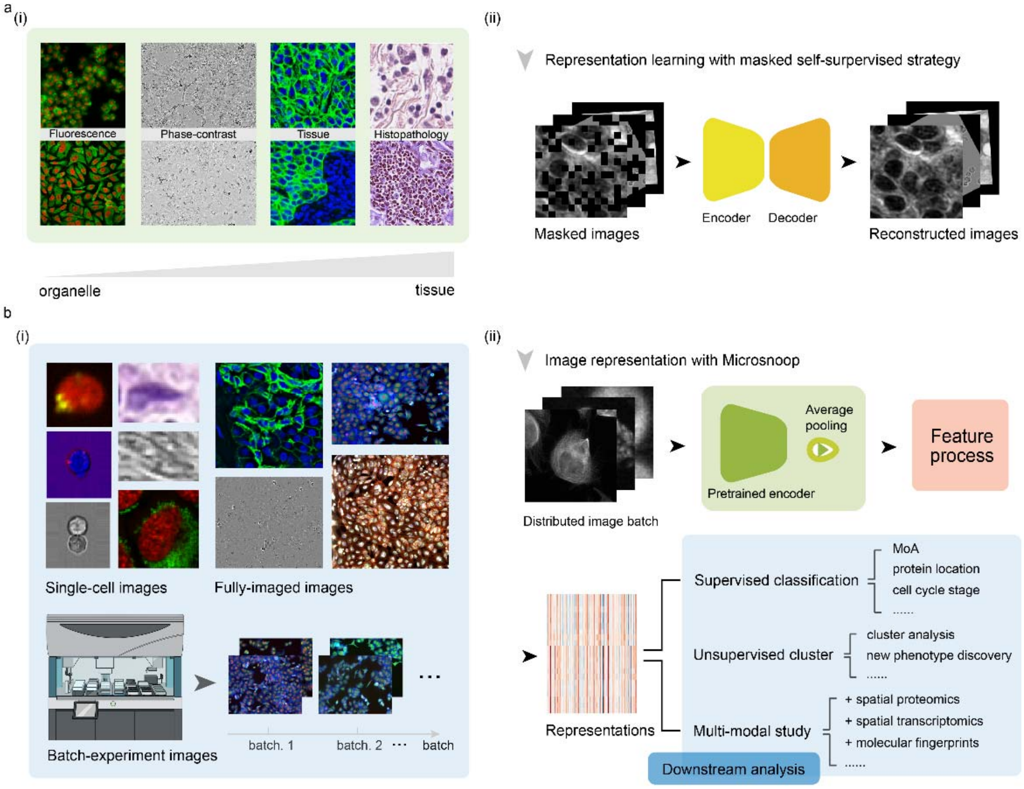


Fig. 1 | Design of Microsnoop for microscopy image representation. a, Schematic of the learning process. (i) Example of the four main category images are shown. The channels range from cellular organelles to tissues. (ii) A masked self-supervised learning strategy was employed and only images are required for training without additional manual annotation. One-channel masked images were set as the input and the Encoder-Decoder were required to reconstruct the original images. **b,** At test time, (i) Example images from various downstream tasks are shown, with different resolutions, number of channels and image types. These microscopy images are categorized into 3 types to ensure the broad coverage of image profiling needs. (ii) Application of Microsnoop. Firstly, images are managed by an in-built task distribution module (Fig. 3a), which generates one batch one-channel images for feature extraction. Each batch of images is fed into the pre-trained encoder, and the output smallest convolutional maps are processed by average pooling. Then, all extracted embeddings are processed according to different profiling tasks (introduced in the following section). The potential downstream analyses of our generalist representation tool are shown in the panel.

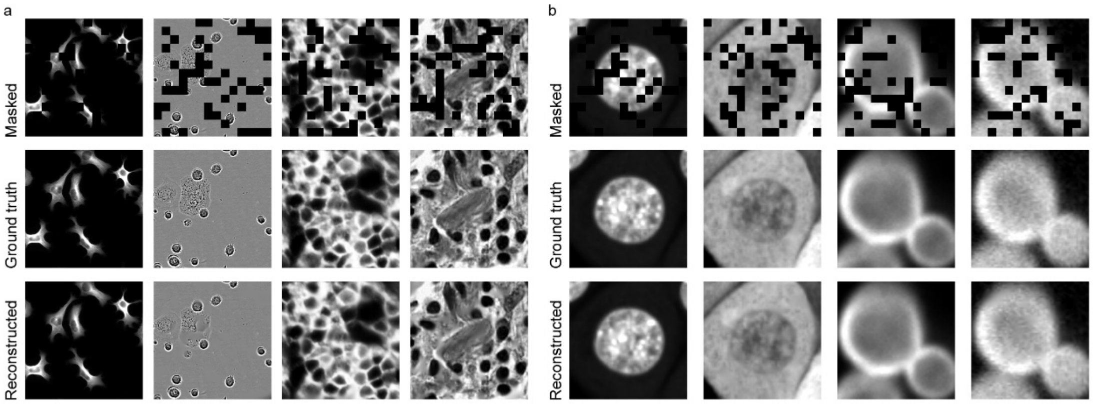


Fig. 2 | Reconstruction results with Microsnoop. a, Example results for images from the validation set, with a masking ratio of 25% applied on inputs. One representative image is selected for each image type. **b,** Example results for single-cell images from evaluated data, with a masking ratio of 25% applied on inputs. The left two columns are from COOS7 and the right two are from CYCLoPs. Two representative images (different imaging channels of the same cell) are selected for each dataset. Example results on other evaluated datasets are shown in Extended Data Figs. 4.

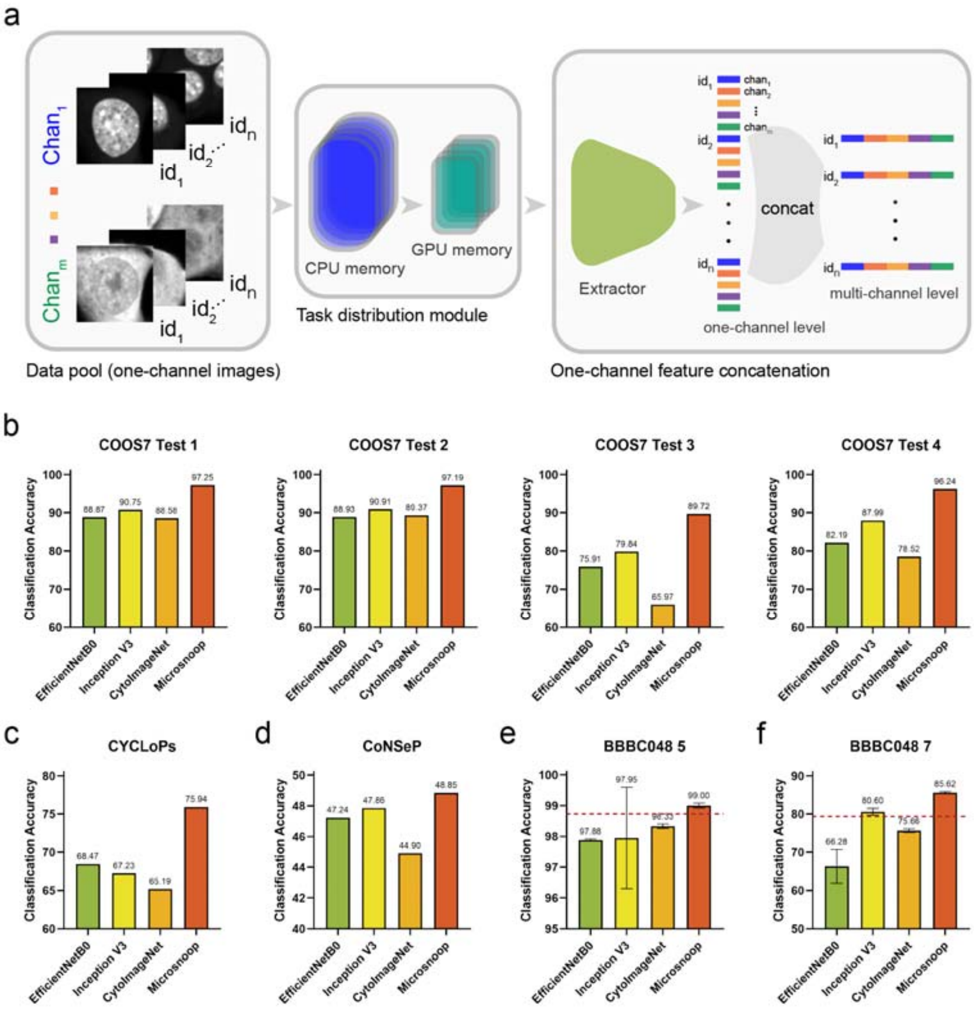


Fig. 3 | Profiling with Microsnoop on single-cell images. a, Pipeline. Every channel of the single-cell image is processed independently, and the one-channel level embeddings are concatenated to get multi-channel level image representations. A task distribution module is provided to prevent memory overflow. The Extractor denotes the pretrained encoder combined with the average pooling layer shown in Fig. 1a(ii). **b-f,** Benchmarks. **b,** Benchmark on COOS7, containing four separate test sets. **c,** Benchmark on CYCLoPs. **d,** Benchmark on CoNSeP. **e,f,** Benchmarks on BBBC048, with two different classification tasks. Performances reported by the original paper are shown with dotted red lines. Error bars represent the mean ± SD of fivefold cross-validation results.

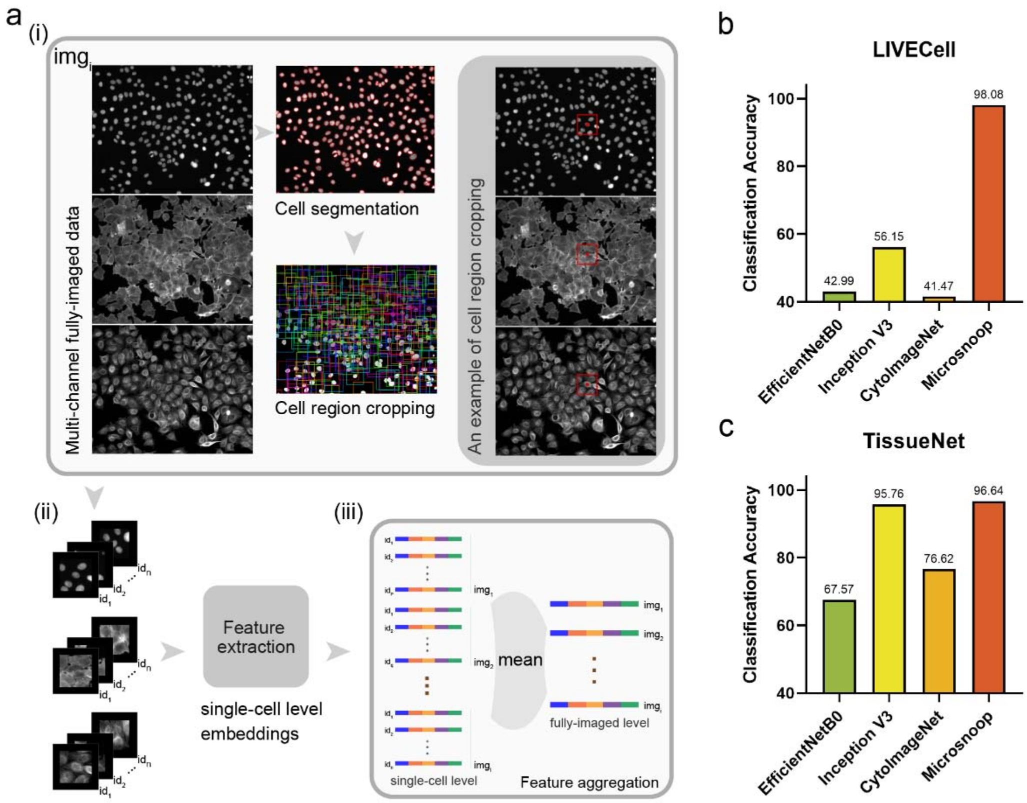


Fig. 4 | Profiling with Microsnoop on fully-imaged images. a, Pipeline. (i) Cell segmentation algorithm is conducted on the easiest channel (such as the nucleus channel) of the multi-channel fully-imaged image, then the cell region for each single cell is computed and cropped. (ii) Multi-channel single-cell images are processed as Fig. 3a, and (iii) the output single-cell level embeddings are aggregated to obtain the fully-imaged level image representations. **b,** Benchmark on LIVECell. **c,** Benchmark on TissueNet.

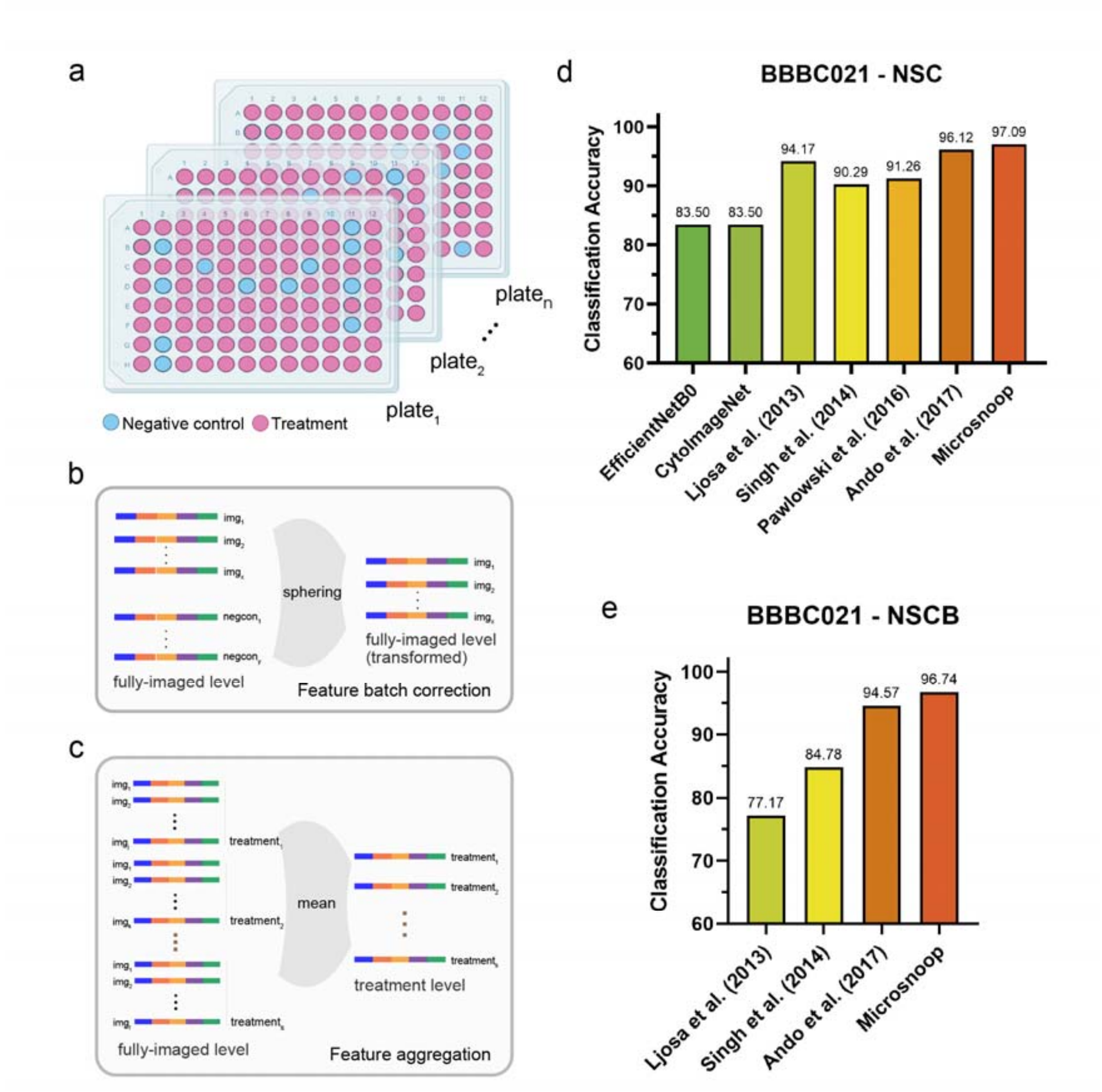


Fig. 5 | Profiling with Microsnoop on batch-experiment images. a, Schematic of multi-well plates in a drug screening experiment containing negative control wells and different treatment wells set in each plate. **b,** Batch correction on fully-imaged level representations. **c,** Feature aggregation on fully-imaged level embeddings to obtain treatment level image representations. **d,e,** Benchmark on BBBC021, with different evaluation metrics.

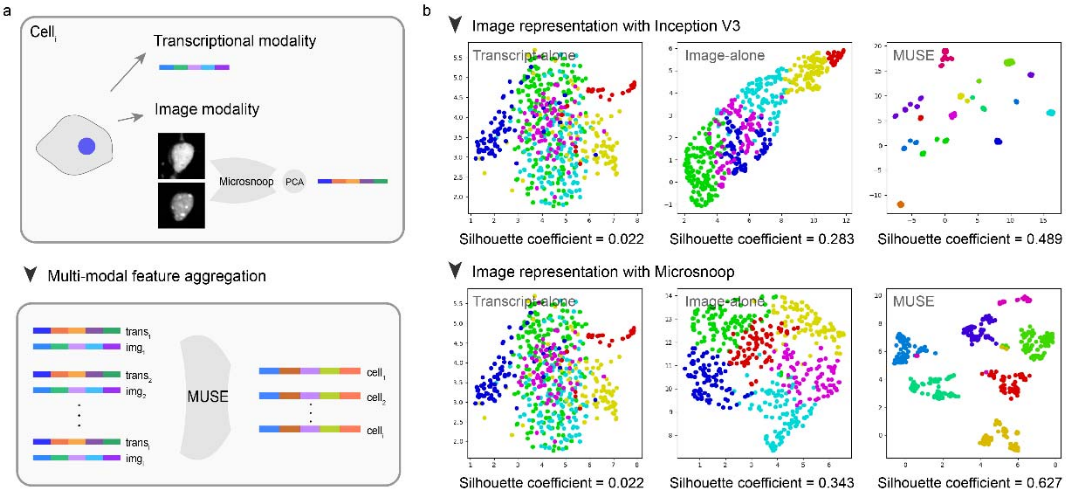
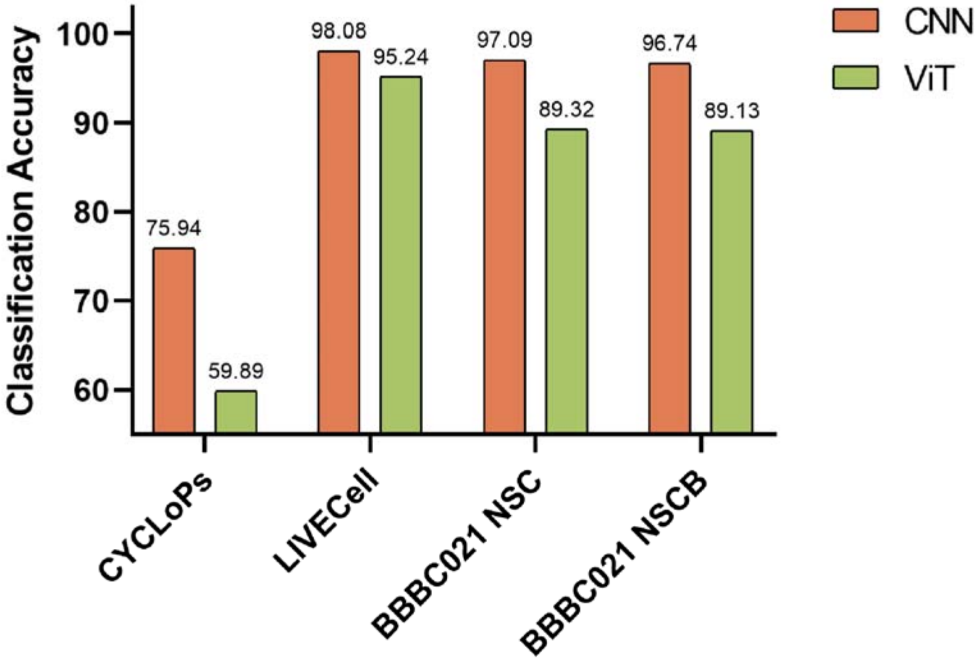
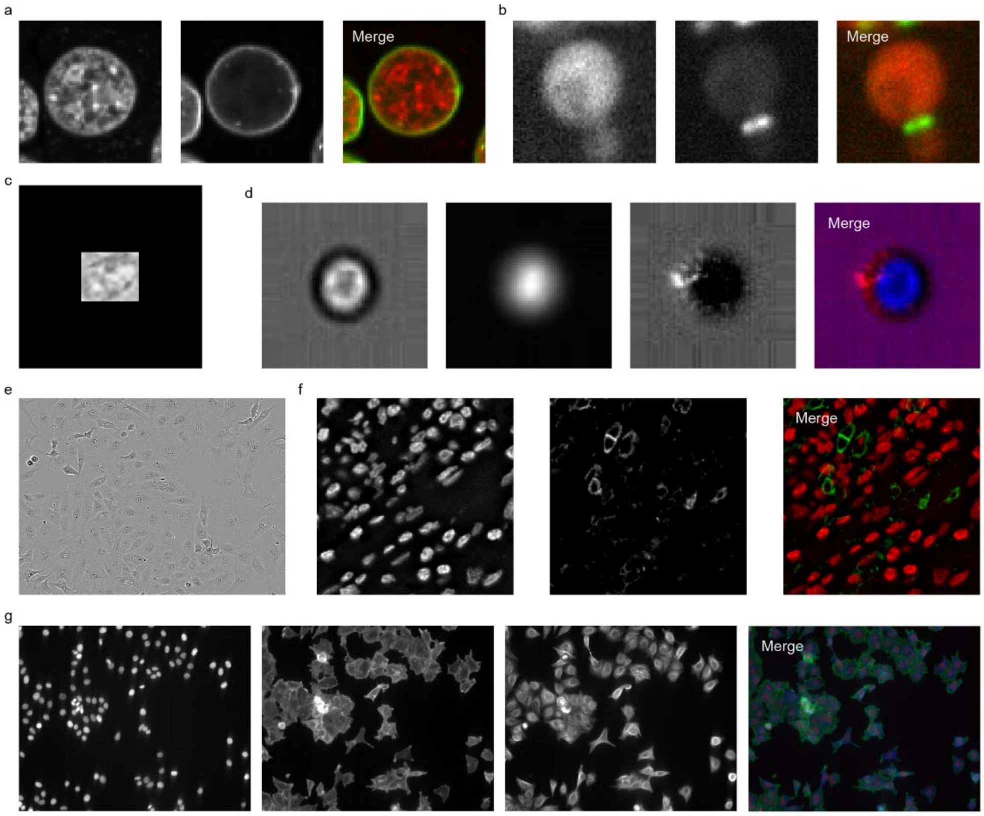


Fig. 6 | Joint use of Microsnoop and MUSE. a, Pipeline. Image modality data is first processed by Microsnoop, then PCA is performed on the output representations to reduce feature dimensionality. Finally, two modality representations are mixed by MUSE. **b,** UMAP visualization of different modality latent spaces on seqFISH+, using two different image representation methods. Silhouette score was used to quantify the separateness of clusters.

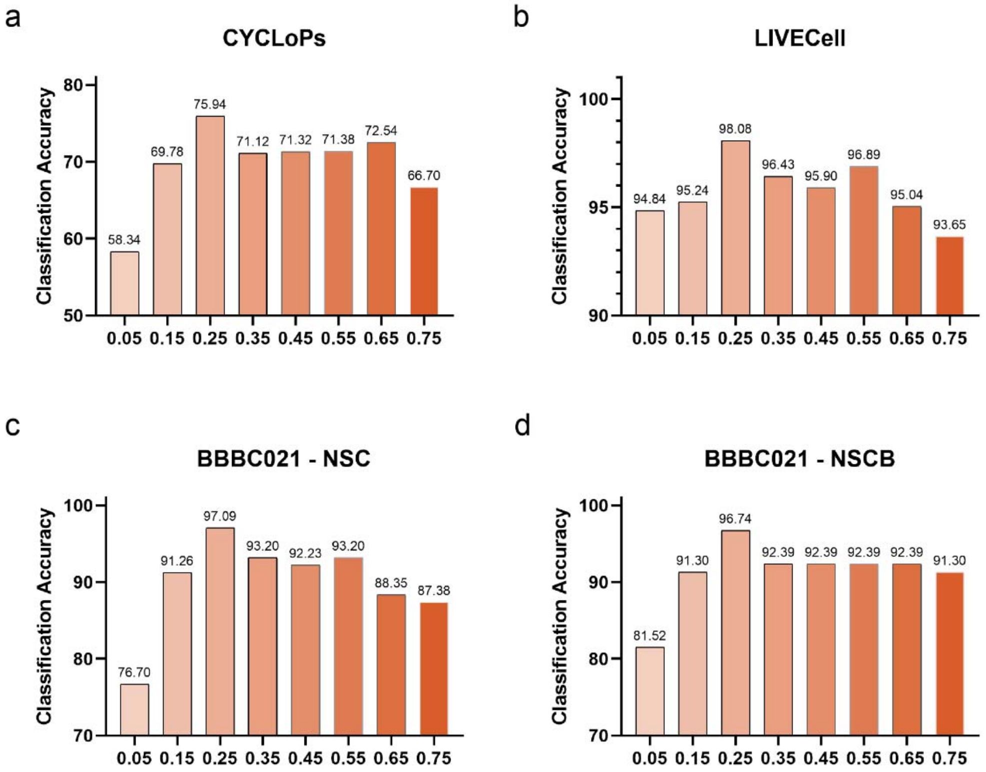
Extended Data



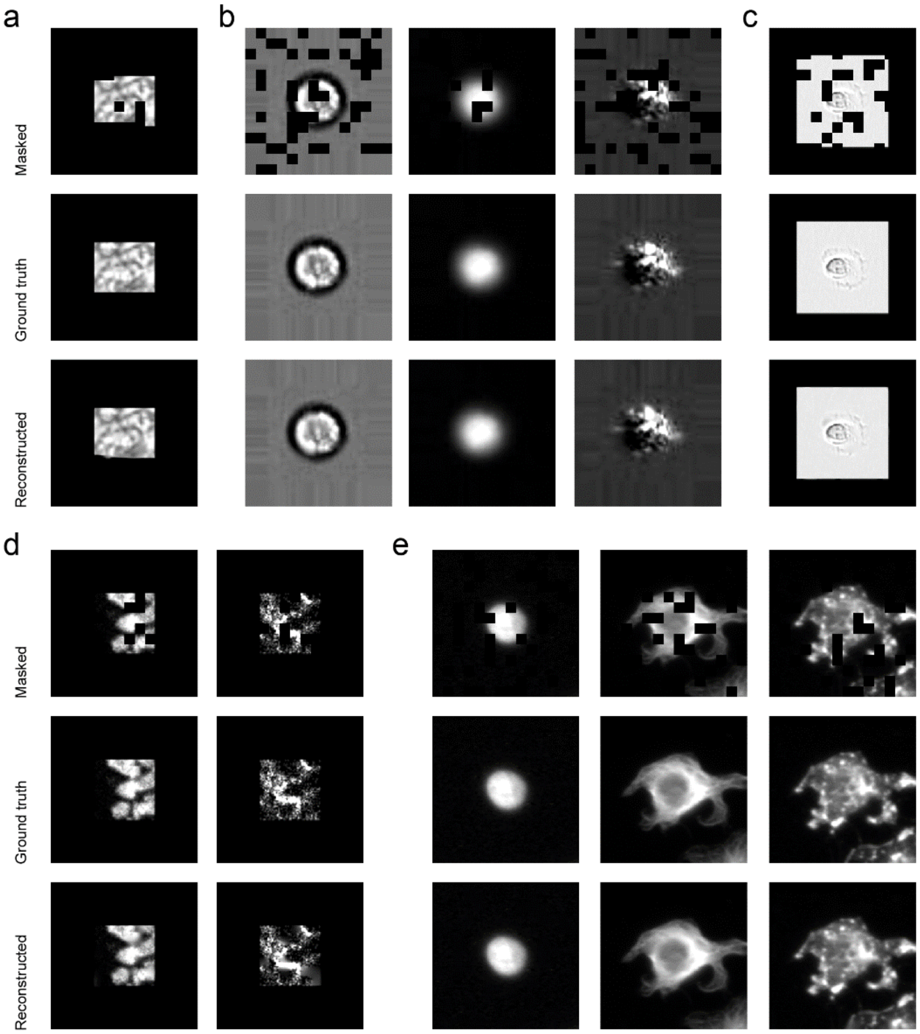
Extended Data Fig. 1 | Performance evaluation of Microsnoop trained with different network architectures. Three representative datasets from seven evaluation datasets were selected for the early trials: single-cell image task (CYCLoPs), fully-imaged image task (LIVECell), and batch-experiment image task (BBBC021). The ViT architecture referred to the MAE, and the classification accuracy for the corresponding dataset was reported.



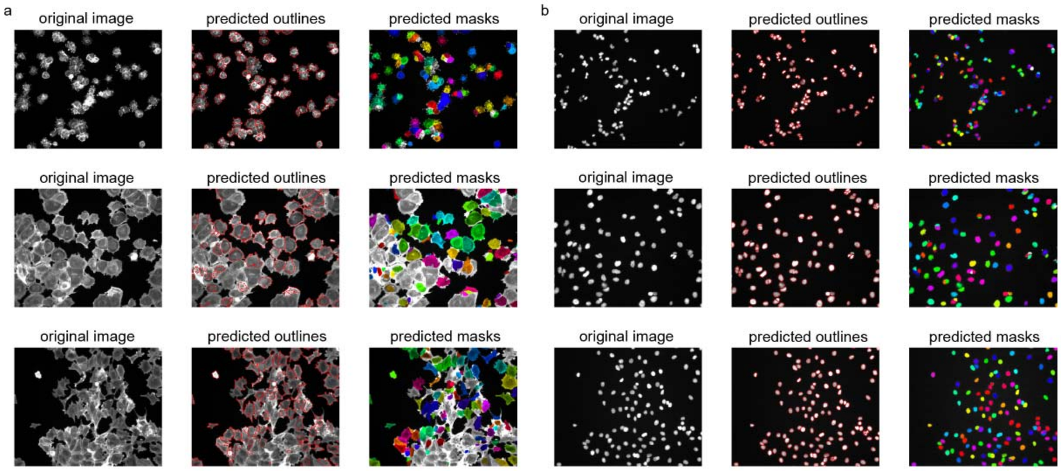
Extended Data Fig. 2 | Example images of evaluation datasets. Each channel of the example image was presented for each dataset: **a,** COOS7 **b,** CYCLoPs **c,** CoNSeP **d,** BBBC048 **e,** LIVECell **f,** TissueNet **g,** BBBC021.



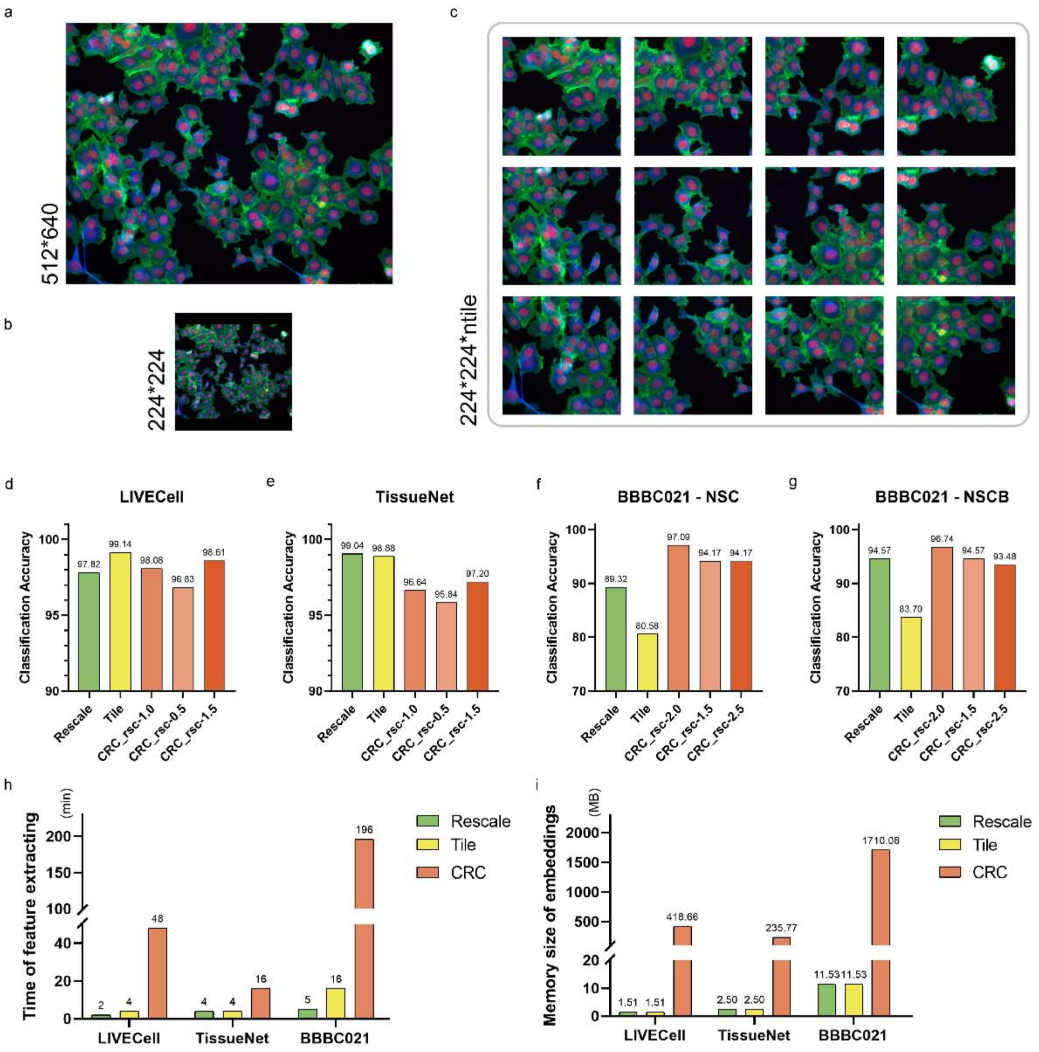
Extended Data Fig. 3 | Performance evaluation of Microsnoop trained with different mask ratios. Three representative datasets from seven evaluation datasets were selected for the early trials: **a,** Single-cell image task **b,** Fully-imaged image task **c,d,** Batch-experiment image task. The mask ratio was set ranging from 0.05 to 0.75, and the classification accuracy for the corresponding dataset was reported.



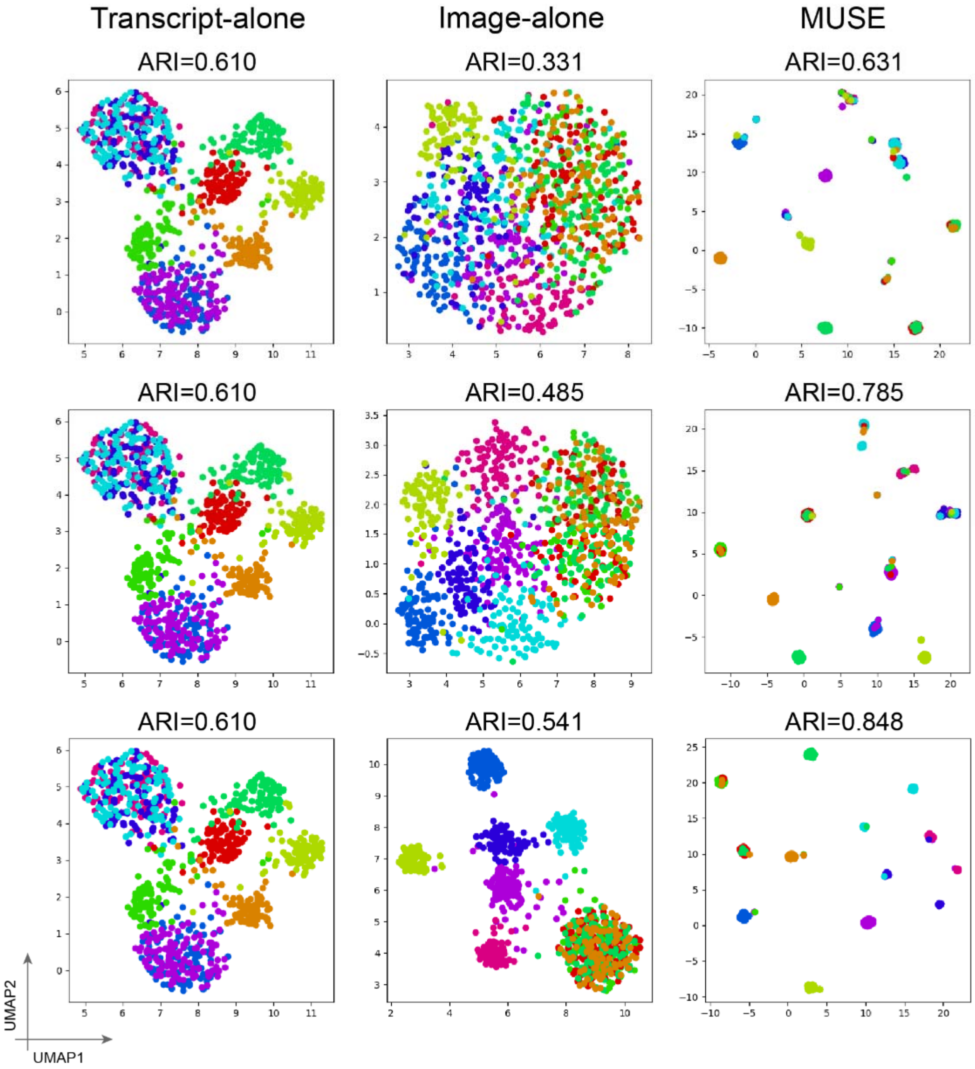
Extended Data Fig. 4 | Reconstruction results with Microsnoop on the remaining evaluation datasets. Each channel of the example images from each dataset were performed: **a,** CoNSeP **b,** BBBC048 **c,** LIVECell **d,** TissueNet **e,** BBBC021. For fully-imaged image datasets (**c-e**), the processed single-cell images after cell region cropping were used.



Extended Data Fig. 5 | Example segmentation results of the generalist model for high-content screening images. Images were shown in pairs, with the original image on the left and the segmentation results on the right using two visualization methods; the predicted outlines show the boundary of each cell and the predicted masks mark the segmented cells with different colors. Three images were selected from the BBBC021 dataset, in which cells were treated with different compounds and presented complex phenotypes. Cell segmentation was conducted with Cellpose. **a,** Segmentation on F-actin channel images. **b,** Segmentation on corresponding nucleus channel images.



Extended Data Fig. 6 | Different profile modes of fully-imaged images. a, An example image. **b,** Example of the rescaling mode, where the original image was patched and rescaled to the input size (224*224). **c,** Example of the tile mode, where the original image is cropped to many 224*224 tiles (ntile) using the make_tiles function from the cellpose.transforms Python package, and the tile_overlap parameter was set as 0.1. **d-g,** Performance comparison of different modes on three evaluation datasets: **d,** LIVECell **e,** TissueNet **f,g,** BBBC021. The cell region cropping mode (CRC) was tested with different rescale constant to study the robustness. **h,i,** Time (**h**) and memory (**i**) cost of different modes. In the case of CRC mode, the memory cost computes the representations of all single-cell images, rather than the final fully-imaged level image representation.



Extended Data Fig. 7 | UMAP visualizations of latent embeddings from single- and combined-modality methods. Colors: ground truth subpopulation labels in simulation. Cluster accuracy is quantified using the adjusted Rand index (ARI).