Cell-APP: A generalizable method for microscopic cell annotation, segmentation, and classification

Anish Virdi[1], Ajit P. Joglekar[1,2,*]

1 – Department of Biophysics, University of Michigan

2 – Cell & Developmental Biology, University of Michigan Medical School

* - ajitj@umich.edu

University of Michigan, Ann Arbor Michigan 48109, USA

Abstract

High throughput fluorescence microscopy is an essential tool in systems biological studies of eukaryotic cells. Its power can be fully realized when all cells in a field of view and the entire time series can be accurately localized and quantified. These tasks can be mapped to the common paradigm in computer vision: instance segmentation. Recently, supervised deep learning-based methods have become state-of-the-art for cellular instance segmentation. However, these methods require large amounts of high-quality training data. This requirement challenges our ability to train increasingly performant object detectors due to the limited availability of annotated training data, which is typically assembled via laborious hand annotation. Here, we present a generalizable method for generating large instance segmentation training datasets for tissue-culture cells in transmitted light microscopy images. We use datasets created by this method to train vision transformer (ViT) based Mask-RCNNs (Region-based Convolutional Neural Networks) that produce instance segmentations wherein cells are classified as "m-phase" (dividing) or "interphase" (non-dividing). While training these models, we also address the dataset class imbalance between m-phase and interphase cell annotations, which arises for biological reasons, using probabilistically weighted loss functions and partisan training data collection methods. We demonstrate the validity of these approaches by producing highly accurate object detectors that can serve as general tools for the segmentation and classification of morphologically diverse cells. Since the methodology depends only on generic cellular features, we hypothesize that it can be further generalized to most adherent tissue culture cell lines.

Introduction

High throughput microscopy and tracking of live cells is an essential approach in cell-biological research, especially in systems biology studies, where accurate quantitation of cell behavior is necessary. Rigorous analysis of this behavior may require the quantification of thousands of individual cells, a tedious task to complete by hand. For this reason, automated image analysis pipelines that accurately track and quantify single-cell state dynamics are necessary. These motivating factors have led to the development of many diverse image analysis methods, including those based on deep learning that localize regions of an image corresponding to individual cells(Schwendy et al., 2020; Jacquemet et al., 2020; Cohen and Uhlmann, 2021; Patel et al.; Fishman et al., 2021; Falk et al., 2019). These methods can be grouped into three broad approaches: customizable machine-learning models, such as Ilastik; Neural networks that categorize image pixels or predict object representations from pixel values, such as CellPose and StarDist, and Neural networks that perform instance segmentation on transmitted light (TL) images, such as LIVECell(Berg et al., 2019; Schmidt et al., 2018; Stringer et al., 2021; Edlund et al., 2021; Ling et al., 2020; Rivenson et al., 2019;

Carpenter et al., 2006), to name a few. These methods have transformed image processing in high throughput microscopy.

Challenges remain, however, in the segmentation of individual cells within fluorescence images. One such challenge typically arises in time-lapse microscopy of genome-edited cells expressing one or more fluorescently labeled proteins. The goal of these studies is to correlate temporal dynamics of the fluorescently tagged protein with cell state changes. Here, segmentation challenges may occur as the endogenously tagged proteins are often expressed at low concentrations and can even disappear at times, e.g., Cyclin B, which is degraded completely at the end of mitosis(Clute and Pines, 1999). Moreover, to mitigate phototoxicity image acquisition is restricted to the low signal-to-noise regime, and photobleaching progressively deteriorates image quality during a time-lapse experiment. These issues can limit the effectiveness of image segmentation methods applied to fluorescence images. Elegant computational methods to recover image quality are being developed to solve these issues (Stringer and Pachitariu, 2024; Krull et al.). Another solution could be to dedicate a fluorescence channel solely for image segmentation, however, this comes at the cost of added manipulation of the cells and leaves one fewer channel for visualizing biological variables.

These problems can be circumvented by instead segmenting the TL images directly as they do not suffer from the aforementioned issues. Additionally, these images contain morphological information about the cell's state, e.g., whether the cell is in interphase or mitosis, regardless of the protein being studied. Just as well, recent developments involving vision transformers have suggested promising new approaches for performing instance segmentation and annotation of label-free cell images(Dosovitskiy et al., 2020). With these considerations in mind, the first aim of this study was to train a vision transformer for the instance segmentation and classification of live tissue culture cells in TL microscopy images. Segmentation of TL images, however, is not without its own challenges. Segmentation across cell lines with diverse morphology, which is less apparent in fluorescence or nuclear images, often proves to be a challenge for current deep-learning methods(Stringer et al., 2021). Datasets such as LIVECell and EVICAN, which contain more than 1.6 million and 26,000 segmented cells across 8 and 30 cell lines, respectively, have aimed to tackle this challenge through the massive manual aggregation of training data(Schwendy et al., 2020; Edlund et al., 2021). Despite their size, neither fully encapsulates the wide array of cellular morphologies found among tissue culture cell lines. The 38 total cell lines of EVICAN and LIVECell constitute only a fraction of the 1,688+ distinct animal cell lines available from American Type Culture Collections, those generated in research labs, and those derived from recent cancer patients. In addition, these datasets do not contain cell-state labels. For these reasons, augmentation of these datasets is necessary for developing improved deep-learning models. Therefore, the second aim of this study was to develop an automated approach to generating instance segmentation training data for tissue culture cells in TL microscopy.

To accomplish these aims, we present Cell-APP (Cellular Annotation and Perception Pipeline), a general automated method for generating instance segmentation training data for tissue culture cells in TL microscopy. Cell-APP utilizes information present within live DNA stain images to assign cell-cycle labels (non-mitotic or "interphase" and mitotic or "m-phase") and generate prompts for Facebook AI Research's SAM (Segment Anything Model), which then produces the mask annotations(Kirillov et al., 2023). Using this approach, we created expansive datasets of HeLa cells in 2D culture and utilized them to train highly accurate vision transformer-based object detectors. These object detectors can then be used to segment transmitted light microscopy images without the need for a fluorescent channel from which to generate prompts. Because Cell-APP's training data generation pipeline has a minimal set of dependencies, namely, visualizable chromatin morphology and apparent intercellular borders, we speculate that this method can be used to rapidly generate training datasets for most adherent cell lines. We evidence this hypothesis by applying this method to U2OS cells and training a similarly accurate object detector. Finally, to facilitate the adoption of this method, we provide a Python package that organizes the application of this method into a succinct pipeline.

Results

Dynamic Cellular Morphology Enables Unsupervised Cell-Cycle State Classification

To circumvent the need for human annotation in the generation of training data, two tasks must be automated: pixel segmentation for each cell in the TL image and classification of the resultant segmentation masks (interphase or m-phase in our case). In most interphase tissue culture cells, chromatin appears as an ovoid of approximately uniform intensity (Fig. 1b). In m-phase cells, however, chromatin compacts significantly before organizing in a plate-like structure known as the metaphase plate. The metaphase plate typically orients orthogonal to the focal plane and appears as a thick line. In anaphase, this thick line splits into two smaller lines or curved segments. Previous studies have shown that chromatin morphology's dependence on cell-cycle state can be used to train a Neural network for classifying live cells(Ulicna et al., 2021). Therefore, we hypothesized that cell-cycle-dependent changes in chromatin morphology, as revealed by a live DNA stain (Silicon-Rhodamine DNA or SiR DNA, Methods), would contain sufficient information for unsupervised classification.

We cropped regions of interest from fluorescence microscopy images of cells treated with SiR-DNA and computed a select subset of geometric and intensity-based properties for each nucleus to produce one vector per region (Methods). For unsupervised classification of this representation, we first reduced the dimensionality of the property vectors to $\mathbb{R}^2$ using Uniform Manifold Approximation and Projection (UMAP) and then used a hierarchical clustering algorithm to partition the low dimensional representation(McInnes et al., 2018; McInnes and Healy, 2017). Two groups arose from the representation: one containing interphase nuclei and the other m-phase (Fig 1c). Re-clustering of the dimension-reduced vectors with only the mitotic cells retained revealed that this cluster could be further segregated to obtain finer-grained cell cycle stage classifications such as anaphase (Methods, Fig, 1c). Over a set of 44 paired TL and SiR-DNA fluorescence images, we used this method to obtain classifications for ~17500 cells with ~75% annotated as interphase and ~ 25% as mitotic.

A closer examination of these data revealed that the cluster labeled as m-phase also contained a small fraction of anaphase cells and dead cells. These could be separated from the true m-phase cells by re-clustering alone. However, with the present dataset, separation of these instances could not be achieved with sufficient recall, i.e., many anaphase and dead cells remained in the m-phase cluster after processing. For this reason, anaphase and dead cell labels were omitted from the final training dataset. Groupings found through this unsupervised pipeline were found by manual inspection to have error rates between 0.05 and 0 (Methods). With a sufficient volume of data, it should be possible to accurately separate and annotate distinct phases of mitosis such as prophase and anaphase for model training. This effort is left for future work.

Promptable Deep Learning Models Automate Image Segmentation

To automate the task of pixel segmentation for each cell in the training dataset, we utilized Facebook AI Research's (FAIR) Segment Anything project, which has produced the foundational "Segment Anything Model" (SAM) and an expansive image segmentation dataset(Kirillov et al., 2023). SAM can segment specific objects in an image that are specified by a human user via a prompt such as a coordinate or bounding box. SAM operates by encoding spatial and textual prompts along with the input image during training and inference, textual prompts are learned through manual provision of mask-text pairs. In generating training data for SAM, FAIR utilized a three-stage program in which a model first trained on common image segmentation datasets was used to generate masks, which were then corrected by the annotator. Subsequent stages required decreasing amounts of supervision by said annotator and the final stage generated training data automatically(Kirillov et al., 2023).

In line with this final stage of data compilation, we provided the XY coordinates of the centroid of each nucleus in the SiR-DNA image to SAM as a prompt for segmenting the corresponding TL image (Fig. 1a, lower right). For segmentation, we utilized SAM models that had been fine-tuned on biological data through the microSAM project(Archit et al., 2023). Worth noting is that this methodology relies on the fact that (1) the centroid of a cell's nucleus approximates the center of the cell, and (2) nuclei rarely overlap. The first dependence is mild, as SAM is robust to imperfect prompting, the second can be circumvented by ensuring that the imaged cells are growing as a monolayer and that the plate being imaged is $< 100\%$ confluent. We also provide support for the usage of bounding boxes as prompts for SAM, where the bounding box corresponds with the coordinates of the region cropped for classification. To validate the usage of SAM for data annotation, we generated prompts using EVICAN dataset annotations as surrogate nuclear fluorescent images and fed those prompts to SAM to generate independent annotations, i.e. we applied the Cell-APP method (Methods). An analysis of 100 images revealed that the average intersection-over-union (IOU) between SAM and EVICAN annotations was 57% (Fig. S1). Therefore, we concluded that the SAM-generated segmentation of individual cells could be used to generate instance segmentation training datasets.

Dataset

The Cell-APP HeLa dataset consists of 32 TL images of cells obtained with a Nikon 20x, 0.46 NA Objective, each annotated through the pipeline consisting of unsupervised classification and SAM prompting, as discussed above. These images provide a total of ~13500 cells, with ~90% annotated as interphase and ~ 10% as m-phase. Images in the datasets vary in cell confluency from < 0.10 to > 0.90 (confluence is informally defined as the fraction of the total area of the substrate occupied by cells), these confluency metrics correlate with the maximum and minimum number of annotated cells per image in the dataset, at 1147 and 26 respectively. Cells in this dataset range in size from 26448 to 770 μm^2 in area (Fig. S2).

Training of Vision Transformers on Cell-APP Annotations Results in Performant Object Detectors

A desirable quality of an object detector is its generalizability: a cell detector trained on the HeLa cell dataset should also be effective in segmenting other cell lines. Outside of cell biology, researchers similarly desire object detectors to accurately detect instances from previously unseen backgrounds or corrupted and blurry images via a process known as out-of-distribution (OOD) generalization. Recent investigations of the generalizability of vision transformers show that their detection performance is less dependent on image background and less hindered by corruption or blur than their CNN counterparts(Zhang et al., 2021; Caron et al., 2021). For this reason, we selected FAIR's Mask-RCNN platform, Detectron2, in conjunction with the plain transformer backbones implemented by Li et al. for training on the Cell-APP generated dataset (Li Hanzi Mao Ross Girshick et al.; Yuxin, 2019). We also selected a foundational convolution-based model (ResNet-50-FPN) for comparison. Experiments were performed using ViT-large, ViT-base, and ResNet50-FPN backboned Mask-RCNNs, each of which was pre-trained on ImageNet-1k. Models were trained via stochastic gradient descent with a total mini-batch size of 2, 4, and 4, respectively. Models were trained for 3000 epochs (Fig. 2a). Backbone comparison using a HeLa cell dataset revealed that both ViT backbones outperform their ResNet counterpart, with mean Average Precisions (mAP) of 55.00, 54.47, and 52.21 for the ViT-large, ViT-base, and ResNet models, respectively (Table 1, Fig. 2a).

Evaluation of the trained model performance showed that the model's mAP in detecting m-phase cells was significantly lower than for detecting interphase cells. We reasoned that this discrepancy in model performance likely arose from class imbalance between m-phase and interphase annotations in our training data, ascribable to the biological reasons noted earlier. To mitigate this issue, we implemented two solutions. First, we experimentally increased the representation of m-phase cells by acquiring images of cells arrested in m-phase-like conditions using the anti-mitotic drug GSK923295. GSK923295 has no known effect on interphase cells, and it does not detectably affect the morphology of the mitotic cells in TL images(Bennett et al.). We refer to this supplemented dataset as HeLa(+), and to the original as HeLa (also denoted by the subscripted + sign in Fig. 2b). The HeLa(+) dataset consists of the original 32 image HeLa dataset, plus an extra 12 TL images, all taken with a 20x objective. The dataset contains ~17500 annotations, with ~75% classified as interphase and ~25% as m-phase.

To evaluate the effect of this additional data on model performance, we trained models on both datasets while keeping the testing and validation datasets unchanged. We utilize F1-score as the measure of model performance as it jointly accounts for changes in mAP and mean Average Recall (mAR) Methods). As seen in Figure 2b, this supplement alone did not improve model performance. Therefore, as a second solution, we modified the prioritization of instances in the loss function during model training. To do this, we utilized a confidence-weighted loss function, "categorical focal loss," as the Mask-RCNN class loss function (Methods). As detailed by Lin et al., focal loss is a variant of categorical cross entropy that reduces the magnitude of an instance's contribution to the model's overall loss in proportion to the "ease" at which that instance was classified(Lin et al., 2017). In other words, individual loss contributions are weighted in inverse accordance with the predicted probability of that instance being the correct class.

Implementation of focal loss in conjunction with the use of supplemented data during model training notably improved model performance (Fig. 2b). In particular, we find that for models trained on the HeLa(+) dataset, focal loss improves F1-score on m-phase annotations by 1.65 and on interphase predictions by 10.32 (Fig. 2b). Interestingly, median prediction confidence is higher among interphase cells as compared to those in m-phase (Fig. S3). Thus, we conclude that Cell-APP generated datasets are indeed sufficient to train performant object-detectors, and that m-phase instance supplementation along with the usage of focal loss improves model performance. Models trained on the HeLa datasets, as well as training statistics, are publicly available at https://zenodo.org/records/14632796.

CyclinB1 Abundance Validates Cell APP Predictions

To experimentally validate the binary cell state classification generated by Cell-APP models, we decided to compare the model-assigned labels to established biological markers of mitosis. One such marker, widely used in cell biology, is the pre-mitotic rise and post-mitotic fall in the abundance of the protein Cyclin B(Gavet and Pines, 2010b; a). Therefore, we imaged genome-edited HeLa cells expressing mNeonGreen-CyclinB1 for 15 hours, used Cell-APP to segment the TL images, and measured the average fluorescence of mNeonGreen-Cyclin B1 over each predicted cell mask for the time series (Methods). Cells were treated with GSK923295 prior to imaging. M-phase start and stop time points were computed as the two most prominent local maxima of the curvature of the graph of concentration vs. time of mNeonGreen-CylcinB1 vs. time as detailed in (Methods, Fig. 3b).

To compare the measures from Cell-APP and our biochemical indicator, we computed simple differences between the start and end time points predicted by Cell-APP and those predicted by CyclinB1 concentrations, $\overrightarrow{\Delta t} \equiv \left\langle \Delta t_{start},\ \Delta t_{end} \right\rangle = \overrightarrow{t}_{CellAPP} - \overrightarrow{t}_{CyclinB}$. Here, $\langle \Delta t_{start}\ ,\ \Delta t_{end} \rangle$ should be interpreted as delays in Cell-APP's prediction of mitotic start and stop as compared to CyclinB1. With 295 cells analyzed in this manner, we find that the median value of Δt_{start} is +20 minutes, or four time points, whereas the median value of Δt_{end} is +10 minutes, or two time points. Thus, there is a net difference of -10 minutes between the mitotic durations predicted by Cell-APP and the CyclinB1 indicator. We note that statistical outliers seen in Figure 3 result from both the occasional classification of dead cells as mitotic and rare sporadic mitotic predictions from Cell-APP.

The discrepancy between the mitotic start times indicated by CyclinB1 and Cell-APP can be explained as follows. As seen in Fig. 3b, the first curvature maximum corresponds to the start of CyclinB1 expression in HeLa cells. However, it is known that CyclinB1 expression in HeLa cells begins late G2. Therefore, our mitotic start indication using CyclinB1 is inherently early. For this reason, the start time predicted by Cell-APP, which relies on the distinct concomitant change in HeLa cell morphology, is likely more accurate (Fig. 3b, bottom). Δt_{end} is partially explained by the following observations. The second maximum in the mNeonGreen-CyclinB1 curvature corresponds to its degradation in anaphase and accurately reflects anaphase onset. Anaphase and telophase cells, however, are labeled as "mitotic" by Cell-APP as discussed earlier. Thus, Cell-APP's prediction of mitotic end will be delayed in comparison to CyclinB1. In summary, Cell-APP introduces a minor bias in predicting the mitotic duration for each cell. More accurate indicators of mitotic start (e.g., CyclinB localization to chromatin following nuclear envelope breakdown) will be used in the future for a more accurate assessment of the systematic biases in Cell-APP performance.

Evidencing the Generality of Cell-APP via Method Application to U2OS Cells

To assess the generalizability of the Cell-APP pipeline for constructing training datasets, we applied it to images of U2OS cells, which have a distinctly different morphology compared to HeLa cells (Fig. 4a-b). We constructed a dataset from GSK923295-treated U2OS cells consisting of 172 TL images with 9,981 total annotated cells at a ~ 65% interphase to ~ 35% m-phase split. Next, we trained ViT-large backboned object detectors using this dataset, and in following suit from our HeLa dataset training protocol, we ran a comparison of the performance of the two class loss functions (focal vs. cross-entropy loss). As before, focal loss improved mAP and mAR by 1.14 and 4.54 points, respectively.

We then assessed the OOD generalization of our trained detectors by evaluating the top-performing HeLa detector on the U2OS testing dataset and the top-performing U2OS detector on the HeLa testing dataset. We also implemented transfer learning by training a HeLa model on the U2OS dataset using both focal and standard cross-entropy loss. We found that the HeLa model achieved a mAP of 42.47 on the U2OS set, which is 11.3 points worse than the U2OS model. Similarly, the U2OS model achieved a mAP of 26.85 on the HeLa set, which is 13.1 points worse than the HeLa model (Fig. 4d). The focal loss and standard cross-entropy transfer learning detectors achieved mAP scores of 40.49 and 40.68 on the U2OS set and 33.08 and 44.67 on the HeLa set, respectively. Of interest is that model performance on their dataset of origin (HeLa) effectively dwindles after transfer learning, which suggests that there may be an optimal epoch to stop the transfer learning scheme such that a combined metric accounting for the dataset of origin and new dataset performance is maximized. This same phenomenon is observed for U2OS detectors trained on the HeLa set via the transfer learning scheme (Fig. 4d).

Following these observations, we assessed whether jointly optimizing model weights over both datasets, i.e., training a baseline detector on both datasets, may result in greater overall performance than the transfer learning scheme. To investigate, we trained ViT-large backboned object detectors on concatenated HeLa+U2OS datasets and recorded mAP performance on both datasets separately. Reporting on $\text{mAP}_{\text{joint}}$ defined to be the mean of the HeLa set and U2OS set mAP, we found that these focal and standard cross-entropy models outperformed their transfer learning counterparts by 11.095 and 4.185 points, respectively (Fig 4d.).

Now with object detectors trained from two distinct datasets in hand, we assessed their OOD generalization abilities by comparing their segmentations to the hand annotations of LIVECell. Specifically, we used the HeLa(+) $\text{ViTL}_{\text{focal}}$ and U2OS $\text{ViTL}_{\text{focal}}$ models to perform instance segmentations of a subset of the LIVECell test dataset that included one image per cell line and computed the IOU between each model's segmentation and LIVECell's hand annotations. We found an average IOU of 0.71 for the HeLa detector and 0.70 for the U2OS detector across the 8 cell lines (Fig. S4). These detectors also assign cell cycle labels to each segmentation; however, it is not possible to quantify the accuracy of this prediction as ground truth segmentations are not classified by cell cycle in the LIVECell dataset.

Discussion

Cell-APP is a general-purpose method for the annotation of live-cell microscopy data, and it offers a series of production-ready, generalizable cell detectors trained on TL images of HeLa and U2OS cells. The methodology largely excises the need for hand annotation from object detector training pipelines by generating annotations programmatically in a two-step process. The first step extracts information inherently found within DNA stain images, namely cell locations and cell-cycle-based class labels. In the second step, this information is fed to FAIR's SAM, or promptable object detectors in general, which results in the segmentation of cells in TL images. Future directions will aim to allow for fine-grained cell-cycle classification within the dataset generation pipeline and to correct the disparity in network confidences across classes (Fig. S3). Of note is that although we use chromatin morphology to generate two cell-cycle class labels, the principle may be extended to generate finer cell cycle classifications using cell-cycle-dependent fluorescent indicators(Bajar et al., 2016) or even labels unrelated to cell cycle. We recommend that users first attempt segmentation with our pre-trained models before using the methodology to train their own, as current models may generalize to previously unseen cell lines (Fig. S4-S5).

Methods

Time-lapse live-cell imaging

Imaging was carried out on an ImageXpress Nano Automated Imaging System (Molecular Devices). A SOLA Light Engine (Lumencore) served as the source of excitation. Cells were seeded in 96-well, glass-bottom imaging plates (Ibidi). Humidified 5% CO2 was supplied to the environment chamber, which was maintained at 37 C. Cells were stained with 1 μM (SiR-DNA (Cytoskeleton) and treated with GSK923295, as needed, 30 minutes prior to imaging.

Classification and annotation generation pipeline

To ensure a vast majority of cells on a given image were located correctly, images were processed interactively in Python using Sci-kit Image. Datasets for property-based classification were also generated in this manner. Dimension reduction was performed via UMAP(McInnes et al., 2018) with minimum data-point distance set to 0 and 25 nearest neighbors. HDBSCAN(McInnes and Healy, 2017) was utilized for clustering, with the minimum number of data points per cluster set to be a scalar multiple of the number of total data points. Each cell was then assigned the cell cycle label corresponding to the parent cluster. We utilized microSAM, which is fine-tuned to segment cellular images, to generate mask annotations(Kirillov et al., 2023). Examples of classification and annotation generation can be found here: https://github.com/anishjv/cell-AAP/tree/main/notebooks . A Jupyter notebook showcasing method evaluation using the EVICAN dataset, as described in the main text, can also be found at the previous link. The generated annotations were split into training, validation, and testing sub-datasets for model training and validation. We employed approximately 30/70 and 40/60 splits, based on the number of individual instances, between the validation/testing sub-datasets and the training set for the supplemented and un-supplemented datasets, respectively. Since training images were taken via time-lapse microscopy, care was taken to ensure that images in the validation and testing sets were not from the same well-position as images in the training dataset. Final datasets are available as COCO format JSON files, as well as in collections of individual instance masks. The code necessary to replicate this work on additional cell lines is also available at https://github.com/anishjv/cell-AAP/tree/main/cell_AAP/annotation.

Model training and evaluation

The Mask-RCNN architecture was chosen for model training due to its consistent presence amongst stat-of-the-art architectures on the COCO instance segmentation test-dev benchmark, as well as ease of model development through FAIR's Detectron2 object-detection platform. FAIR's Detectron2 platform is implemented programmatically via PyTorch. ViT-L, ViT-B, and ResNet50 backbone architectures were chosen for training experiments performed using the Cell-APP HeLa dataset(Yuxin, 2019). Training was initialized from model weights learned on ImageNet-1K and performed on 1 NVIDIA A40 GPU with a memory size of 48 GB. We trained each model for 3000 epochs, storing weights every 300 epochs and computing performance metrics on the validation dataset every 150 epochs to probe for overfitting. AdamW (Adam with weight decay) is used as the model optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.99$) with stepwise learning rate decay and a linear learning rate warm-up for 250 iterations. The original images (2048x2048 pixels) were down-sampled so that the input image size was 1024x1024. Patch size was set at 16 for the vision transformer models. Images were subjected to an augmentation regime consisting of scaling, cropping, re-orienting, and modifying brightness and contrast. Detailed model parameters and image augmentation specifications along with training logs and weights are publicly available at: https://zenodo.org/me/uploads?q=&l=list&p=1&s=10&sort=newest.

Focal Loss implementation for addressing class imbalance

Binary cross entropy loss, utilized for quantizing model error on a class prediction given two annotation classes, is defined as:

$$CL_{CE} = \begin{cases} -log(p) \text{ if } y = 1 \\ -log(1-p) \text{ otherwise} \end{cases}$$

Here $y \in [0, 1]$ is the instance's true class and $p \in [0, 1]$ is the model's predicted probability that $y = 1$. It is helpful to rewrite p in a way that always represents the model's predicted probability of the instance being of the true class and not simply of class $y = 1$. We do this in the following way, which also modifies our definition of cross-entropy loss.

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases}$$

$$L_{CE} = -log\left(p_t\right)$$

In model training, this loss function is applied across all instances and a mean is taken to determine the model's total class loss. Focal loss aims to downweight the contribution of "easy" (large p_t) instances to the total loss. Lin et al. achieved this by appending a scalar, constructed from the probability of the model predicting the correct class, p_t to the definition of cross-entropy. The scalar takes the form $(1 - p_t)^\gamma$ where $\gamma \geq 0$, termed the focusing parameter, exists to increase the magnitude by which the contribution from "easy" instances are downweighed. Focal loss is defined as:

$$L_{focal} = -\left(1 - p_t\right)^\gamma \log\left(p_t\right)$$

Focal loss, as well as cross-entropy, is easily extended to the multi-class case by summing over L_{focal} or L_{CE} for N predicted probabilities p_n, where N is the number of classes. This value is, in practice, divided by N to yield an average loss over all possible classes. For model training in which focal loss is utilized, we set $\gamma = 2$. Further analysis of focal loss can be found in [20].

Model evaluation and benchmarking

In evaluating model performance, we chiefly follow the standard COCO guidelines. Given the large, often exceeding 1K, number of instances in a given image, we raise the standard maximum number of detections from 100 to 2000. AP and AR were utilized in model comparison and in deciding upon the optimal set of weights to use in subsequent data analysis applications. AP and AR are defined in terms of three fundamental metrics, namely, precision, recall, and intersection over union. precision and recall are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}\text{Recall} = \frac{TP}{TP + FN}$$

Here, TP, FP, and FN refer to true positive, false positive, and false negative respectively. Qualitatively, precision informs on a model's tendency to make correct predictions, and recall informs on the model's ability to predict all instances. These statistics are fundamentally based on intersection-over-union (IOU), which partially defines whether a prediction is considered TP, FP, or FN. IOU is defined as:

$$\text{IOU} = \frac{Prediction \bigcap Target}{Prediction \bigcup Target}$$

IOU represents the overlap between the predicted bounding box or segmentation and the nearest actual bounding box or segmentation. In practice, a threshold, t, is chosen such that if IOU $> t$ the prediction is considered TP if the prediction is of the correct class and FP otherwise. Similarly, if IOU $< t$ the prediction is considered FN. Precision and Recall are computed for each prediction and each class, allowing us to construct new functions $p_c(r)$ representing the precision as a function of recall for each class c. Note that precision and recall are meaningless for individual predictions, in practice, predictions are sorted in descending model confidence, and TP, FP, and FN are updated before each computation of prediction and recall. AP and mAP are then defined as:

$$AP_c = \int p_c(r)\, dr$$

$$mAP = \frac{1}{N_c} \sum_C AP_c$$

Here N_c is the number of classes. In plain terms, AP_C it is the area under the precision vs. recall curve for that specific class. Notice that the value of AP_C it depends on the IOU threshold, t, that we initially picked. In practice, this threshold is often set to 0.5 or 0.75. It may even be defined as a sequence of values, in this case, AP_C is computed for each t , and the average is used as the final result. From COCO, we adopt, and report values based on this definition of t as the sequence of 10 values from 0.5 to 0.95 inclusive. We also compute AP values by object size analogously to the computation of AP_C.

Similarly, mAR is constructed by computing recall for a given class, c, at this same sequence of IOU thresholds, t. mAR for a given class is then defined as follows:

$$AR_c \;=\; \int Recall_c(t) \;\; dt$$

$$mAR \;=\; \frac{1}{N_c} \sum_C AR_c$$

During detector training, mAP and mAR were measured on the respective testing set every 300 epochs, and model weights were saved every 600 epochs. In model comparison experiments, except the cross-cell-line comparison experiments, we report training and evaluation statistics from the epoch of greatest mAP. For the cross-cell-line comparison investigation, we report evaluation statistics procured from the final output model, i.e., the weights as of epoch 3000. This was necessitated by the need to evaluate these detectors on a dataset other than their native train-test dataset and because model weights were recorded every 600 epochs as opposed to every 300. Due to these stipulations, we could not report statistics from the epoch of greatest mAP as the epoch of greatest mAP may have referred to a model of epoch 300n, whereas saved models were of epoch 600n.

F1-score was utilized in evaluating the effect of m-phase annotation supplementation and focal loss on HeLa detector performance. F1-score, defined as:

$$2\frac{Precision \cdot Recall}{Precision + Recall}$$

was chosen for this task for its ability to summarize changes in both precision and recall. For this assay, we computed F1-scores for interphase and m-phase predictions separately using precision and recall metrics taken from the epoch of greatest mAP.

CyclinB1 concentration dynamics

Genome-edited HeLa cells expressing mNeonGreen-CyclinB1 were imaged every 5 minutes. mNeonGreen fluorescence were median filtered, background-subtracted and max-scaled. Background fluorescence intensity was computed by imaging a well containing only DMEM. The temporal mean of this stack was taken as a background mapping of dimension and size equivalent to a single image from any given sample well (N_p, N_p). Analogously to the background mapping, temporal mappings were generated by imaging a well containing only Fluorobrite. Max-scaling was applied temporally to this (N_t, N_p, N_p) Dimensioned stack. Median filtering was applied to each of these mappings to counteract random bright and dark pixels present in the images.

Mean intensity within regions corresponding to each cell, as identified by Cell-AAP, was taken to be measure of intracellular CyclinB1 concentration. Cells were tracked through time using b-track(Ulicna et al., 2021), which allowed for the construction of N discretely valued functions of the form $[CycB1]_N(t_n)$, where N is the number of cells and n is the number of time points. Each function was smoothed using Kernel Density Estimation to allow for the computation of $\frac{d[CycB1]_N}{dt}$ and subsequent derivatives via the method of finite differences.

We used the curvature, $\kappa(t)$, of the smoothed $[CycB1]_N(t_n)$ to determine time points corresponding to the start and end of mitosis. As mentioned previously, these time points were determined by locating the local maxima of $\kappa(t)$. The algebraic form of the curvature of a scalar-valued function, such as $[CycB1]_N(t_n)$, can be derived by considering a parametric plane curve $\overrightarrow{r}(\mathbf{t}) = (\mathbf{x}(\mathbf{t}), \mathbf{y}(\mathbf{t}))$**.** Every plane curve can be parametrized by its arclength, s, which is defined as:

$$s(t) = \int_a^t \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}\, dt$$

When parametrized in terms of arc length, the curve takes the form, $\overrightarrow{r}(\mathbf{s}) = (\mathbf{x}(\mathbf{s}), \mathbf{y}(\mathbf{s}))$, and each step along the curve corresponds to traveling 1 unit of distance. From this parametrization of the plane curve we can construct a unit tangent function, $\overrightarrow{T}(s)$, that is at any point tangent to the function and has length equal to one. We then define curvature as the degree to which $\overrightarrow{T}(s)$ changes as we move along the curve. Formally, $\kappa = \left|\frac{d\overrightarrow{T}}{ds}\right|$. Another way of saying this is that κ is the rate at which the curve changes direction as one walks along the curve.

From this definition, the curvature of a plane curve, $\overrightarrow{r}(\mathbf{t}) = (\mathbf{x}(\mathbf{t}), \mathbf{y}(\mathbf{t}))$**,** can be derived as:

$$\kappa = \frac{x^{'}y^{''} - y^{'}x^{''}}{(x'^2 + y'^2)^{3/2}}$$

Here, prime notation signifies a derivative with respect to t. We can also note that the graph of a function is simply a special case of a plane curve in which $x(t) = t$ and hence $\overrightarrow{r}(\mathbf{t}) = (\mathbf{t}, \mathbf{y}(\mathbf{t}))$. In this case, the curvature reduces to:

$$\kappa = \frac{y^{''}}{(1 + y'^2)^{3/2}}$$

The two most prominent local maxima of κ were then determined through neighborhood value comparison and prominence (vertical distance from baseline) filtering. In computing the difference between the CyclinB1 and Cell-APP predicted mitotic ranges, t values for both peaks were subtracted from mitotic stop and end t values as predicted by Cell-APP.

$$(\Delta t_{start}, \Delta t_{end}) = (t_{start,Cell-APP}, t_{end,Cell-AAP}) - (t_{start,CycB1}, t_{end,CycB1})$$

We find that $(\Delta t_{start}, \Delta t_{end}) > (0,\ 0)$ and so the physical interpretations of $(\Delta t_{start}, \Delta t_{stop})$ are the delay in Cell-APP's prediction of mitotic start and the delay in Cell-AAP's prediction of mitotic end. In the main text, two common outliers of this statistic are described. In the first case, Cell-APP will have called a dead cell mitotic, and in most cases, these dead cells persist within the field of view for the entire time series. If this is the case, Cell-APP will continue to call the dead cell mitotic for the majority of the time series, and the dead cell can hence be removed from the dataset by filtering out true statistical outliers (if the dead cell is called mitotic for the entire time series its time in mitosis will be a statistical outlier). In the case that a cell dies near the end of a time series, its time in mitosis may resemble that of a healthy cell. However, one may remove it from the dataset by filtering out cells that are still in mitosis in the last frame of time-series.

Regarding the second-mentioned case, Cell-APP will infrequently make mitotic predictions of low confidence. If this is true, the cell-in-question's predicted state may "flicker" between m-phase and interphase throughout the time series. These temporal state predictions, represented as a series of 0s and 1s, can be "smoothed" to either all 1(interphase) or all 0(m-phase) via median filtering.

Acknowledgments

We thank Drs. Matthew O'Meara and Idse Heemskerk for insightful comments and suggestions regarding the writing and content of the paper. Anish Virdi thanks Mishan Gagnon for ever inisghtful conversations that provoked him to investigate new methods for automating dataset generation. This work was supported by R35-GM126983 to APJ for NIGMS.

References

Archit, A., S. Nair, N. Khalid, P. Hilt, V. Rajashekar, M. Freitag, S. Gupta, A. Dengel, S. Ahmed, and C. Pape. 2023. Segment Anything for Microscopy. doi:10.1101/2023.08.21.554208.

Bajar, B.T., A.J. Lam, R.K. Badiee, Y.H. Oh, J. Chu, X.X. Zhou, N. Kim, B.B. Kim, M. Chung, A.L. Yablonovitch, B.F. Cruz, K. Kulalert, J.J. Tao, T. Meyer, X.D. Su, and M.Z. Lin. 2016. Fluorescent indicators for simultaneous reporting of all four cell cycle phases. *Nat Methods*. 13:993–996. doi:10.1038/nmeth.4045.

Bennett, A., B. Bechi, A. Tighe, S. Thompson, D.J. Procter, and S.S. Taylor. Cenp-E inhibitor GSK923295: Novel synthetic route and use as a tool to generate aneuploidy. 6.

Berg, S., D. Kutra, T. Kroeger, C.N. Straehle, B.X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, J.I. Cervantes, B. Xu, F. Beuttenmueller, A. Wolny, C. Zhang, U. Koethe, F.A. Hamprecht, and A. Kreshuk. 2019. ilastik: interactive machine learning for (bio)image analysis. *Nat Methods*. 16:1226–1232. doi:10.1038/s41592-019-0582-9.

Caron, M., H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. 2021. Emerging Properties in Self-Supervised Vision Transformers.

Carpenter, A.E., T.R. Jones, M.R. Lamprecht, C. Clarke, I.H. Kang, O. Friman, D.A. Guertin, J.H. Chang, R.A. Lindquist, J. Moffat, P. Golland, and D.M. Sabatini. 2006. CellProfiler: Image analysis software for identifying and quantifying cell phenotypes. *Genome Biol*. 7. doi:10.1186/gb-2006-7-10-r100.

Clute, P., and J. Pines. 1999. Temporal and spatial control of cyclin B1 destruction in metaphase. 1.

Cohen, E., and V. Uhlmann. 2021. Aura-net: Robust segmentation of phase-contrast microscopy images with few annotations. *In* Proceedings - International Symposium on Biomedical Imaging. IEEE Computer Society. 640–644.

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

Edlund, C., T.R. Jackson, N. Khalid, N. Bevan, T. Dale, A. Dengel, S. Ahmed, J. Trygg, and R. Sjögren. 2021. LIVECell—A large-scale dataset for label-free live cell segmentation. *Nat Methods*. 18:1038–1045. doi:10.1038/s41592-021-01249-6.

Falk, T., D. Mai, R. Bensch, Ö. Çiçek, A. Abdulkadir, Y. Marrakchi, A. Böhm, J. Deubner, Z. Jäckel, K. Seiwald, A. Dovzhenko, O. Tietz, C. Dal Bosco, S. Walsh, D. Saltukoglu, T.L. Tay, M. Prinz, K. Palme, M. Simons, I. Diester, T. Brox, and O. Ronneberger. 2019. U-Net: deep learning for cell counting, detection, and morphometry. *Nat Methods*. 16:67–70. doi:10.1038/s41592-018-0261-2.

Fishman, D., S.O. Salumaa, D. Majoral, T. Laasfeld, S. Peel, J. Wildenhain, A. Schreiner, K. Palo, and L. Parts. 2021. Practical segmentation of nuclei in brightfield cell images with neural networks trained on fluorescently labelled samples. *J Microsc*. 284:12–24. doi:10.1111/jmi.13038.

Gavet, O., and J. Pines. 2010a. Progressive Activation of CyclinB1-Cdk1 Coordinates Entry to Mitosis. *Dev Cell*. 18:533–543. doi:10.1016/j.devcel.2010.02.013.

Gavet, O., and J. Pines. 2010b. Activation of cyclin B1-Cdk1 synchronizes events in the nucleus and the cytoplasm at mitosis. *Journal of Cell Biology*. 189:247–259. doi:10.1083/jcb.200909144.

Jacquemet, G., E. Fazeli, N.H. Roy, G. Follain, R.F. Laine, L. von Chamier, P.E. Hänninen, J.E. Eriksson, and J.Y. Tinevez. 2020. Automated cell tracking using StarDist and TrackMate. *F1000Res*. 9. doi:10.12688/f1000research.27019.1.

Kirillov, A., E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. 2023. Segment Anything.

Krull, A., T.-O. Buchholz, and F. Jug. Noise2Void-Learning Denoising from Single Noisy Images.

Li Hanzi Mao Ross Girshick, Y., K. He, and equal contribution. Exploring Plain Vision Transformer Backbones for Object Detection.

Lin, T.-Y., P. Goyal, R. Girshick, K. He, and P. Dollár. 2017. Focal Loss for Dense Object Detection.

Ling, C., M. Halter, A. Plant, M. Majurski, J. Stinson, and J. Chalfoun. 2020. Analyzing U-net robustness for single cell nucleus segmentation from phase contrast images. *In* IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE Computer Society. 4157–4163.

McInnes, L., and J. Healy. 2017. Accelerated Hierarchical Density Clustering. doi:10.1109/ICDMW.2017.12.

McInnes, L., J. Healy, and J. Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.

Patel, G., H. Tekchandani, and S. Verma. Cellular Segmentation of Bright-field Absorbance Images Using Residual U-Net.

Rivenson, Y., T. Liu, Z. Wei, Y. Zhang, K. de Haan, and A. Ozcan. 2019. PhaseStain: the digital staining of label-free quantitative phase microscopy images using deep learning. *Light Sci Appl*. 8. doi:10.1038/s41377-019-0129-y.

Schmidt, U., M. Weigert, C. Broaddus, and G. Myers. 2018. Cell Detection with Star-convex Polygons. doi:10.1007/978-3-030-00934-2_30.

Schwendy, M., R.E. Unger, and S.H. Parekh. 2020. EVICAN - A balanced dataset for algorithm development in cell and nucleus segmentation. *Bioinformatics*. 36:3863–3870. doi:10.1093/bioinformatics/btaa225.

Stringer, C., and M. Pachitariu. 2024. Cellpose3: one-click image restoration for improved cellular segmentation. doi:10.1101/2024.02.10.579780.

Stringer, C., T. Wang, M. Michaelos, and M. Pachitariu. 2021. Cellpose: a generalist algorithm for cellular segmentation. *Nat Methods*. 18:100–106. doi:10.1038/s41592-020-01018-x.

Ulicna, K., G. Vallardi, G. Charras, and A.R. Lowe. 2021. Automated Deep Lineage Tree Analysis Using a Bayesian Single Cell Tracking Approach. *Front Comput Sci*. 3. doi:10.3389/fcomp.2021.734559.

Yuxin, W. and A.K. and F.M.W.-Y.L. and R.G. 2019. Detectron2.

Zhang, C., M. Zhang, S. Zhang, D. Jin, Q. Zhou, Z. Cai, H. Zhao, X. Liu, and Z. Liu. 2021. Delving Deep into the Generalization of Vision Transformers under Distribution Shifts.

Figures

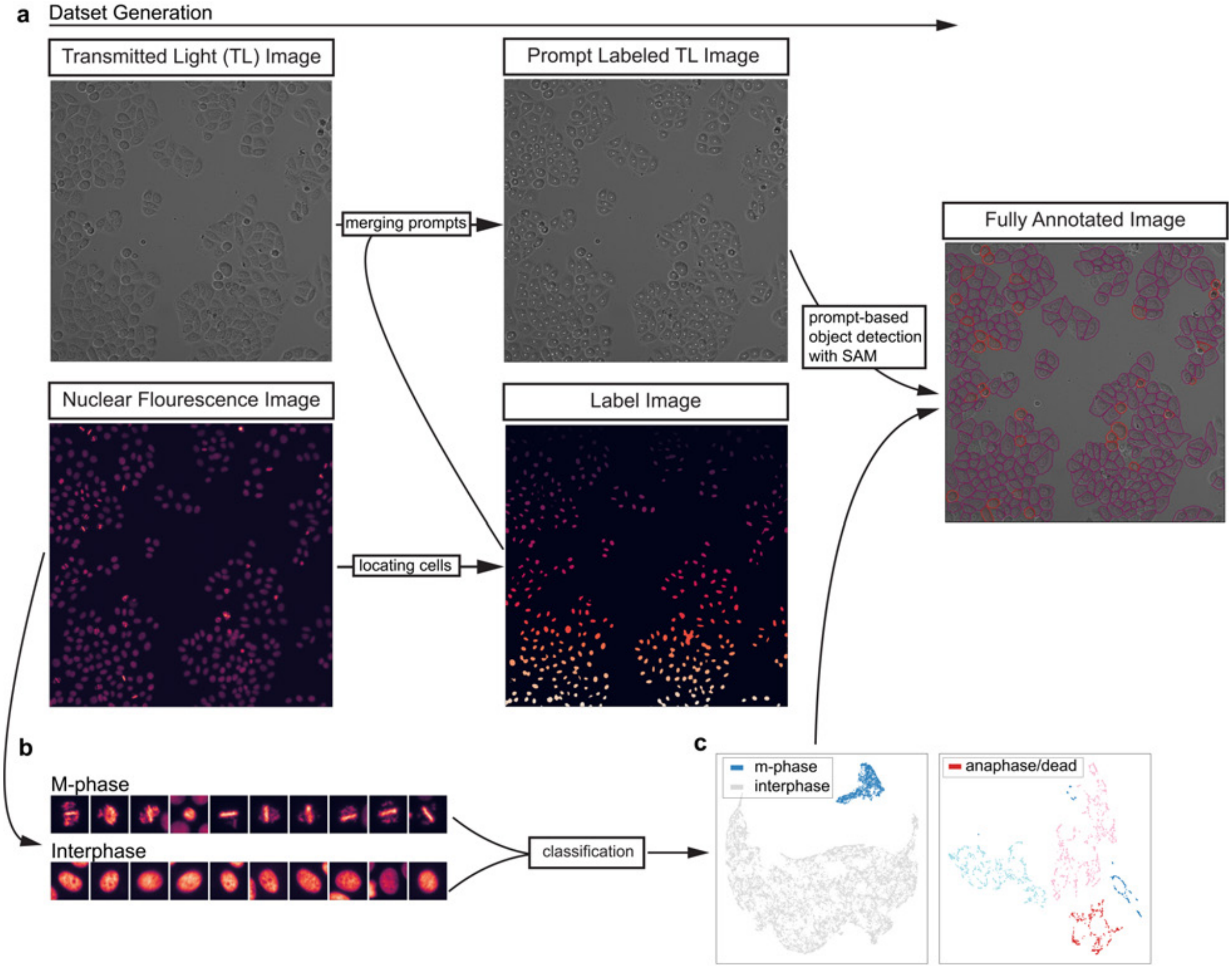

Figure 1: Unsupervised Classification and Segmentation Generate Annotations of Sufficient Quality to Train an Object-Detector

a) Overview of the Cell-APP annotation/dataset generation pipeline. First, cells are localized from a nuclear fluorescent image (bottom micrograph: cells stained with SiR-DNA). Prompts, e.g., bounding boxes and/or centroids, are extracted from these localizations (label image). Prompts are then overlaid on the transmitted light image (top micrograph) and fed to S.A.M for mask generation.

b) Cropped exemplary regions of interest displaying mitotic (top) and interphase cells (bottom).

c) (right) UMAP metric from which the clusters displayed in panel b emerge. (left) UMAP metric of the isolated m-phase cluster. Metrics were clustered using HDBSCAN. The two unlabeled clusters in the isolated m-phase metric do not consist of any one cell type, such as prophase, metaphase, etc. Each point represents the UMAP projection of an N-dimensional vector $\mathbb{R}^2$.

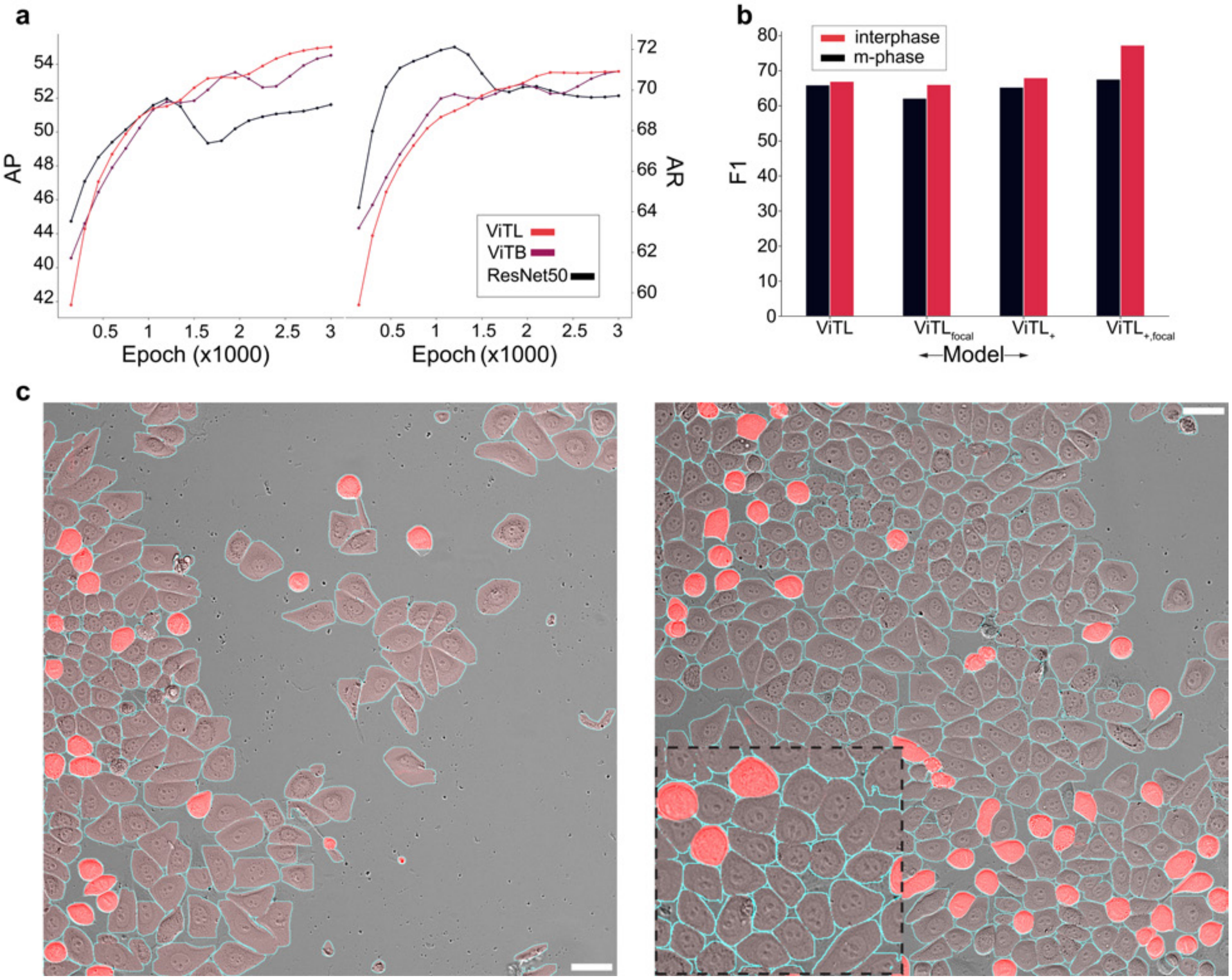

Figure 2: Object-Detector Training Confirms Dataset Sufficiency and Cell-APP Method Validity

a) Mean-average precision and recall vs. training epoch curves. Average precision and recall were measured from the respective network's bounding box predictions.

b) Comparison of F1-score across various datasets and loss function combinations. Subscript (+) denotes the addition of m-phase cell-heavy images to the training dataset (Methods). Subscript *"focal"* denotes the use of focal cross entropy loss in place of standard cross entropy loss for model optimization (Methods).

c) Example inference results on low- and high-density images of HeLa cells taken with a 20x objective. Bright red denotes an m-phase prediction, while dim red denotes an interphase prediction. Cell boundaries are given in cyan. Scale bars are roughly 60 microns in length.

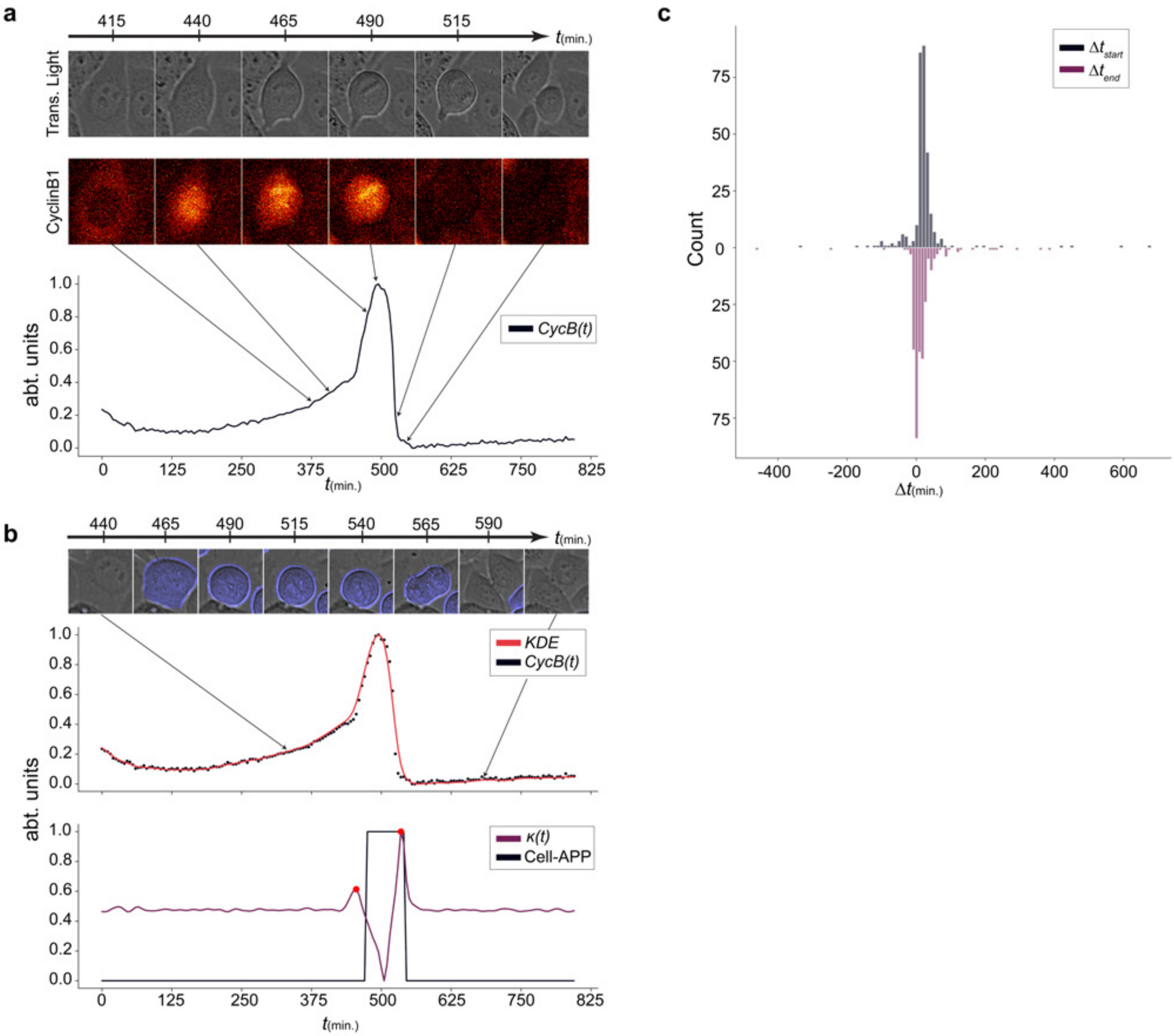

Figure3: Validation of Cell-APP cell state predictions via quantification of cellular Cyclin B1 abundance.

a) Correspondence of CyclinB1 rise and visual changes in cellular morphology. (Top) TL microscopy images, (middle) fluorescent images of the same cell indicating the intracellular concentration of CyclinB1, and (bottom) the corresponding smoothed mNeonGreen-CyclinB1 signal as a function of time.

b) (Top) HeLa cell undergoing mitosis where blue shading indicated that Cell-APP had called the cell “mitotic”. The following graphs correspond to the pictured cell. (Middle), KDE smoothing of the graph of CyclinB1 concentration vs. time, and (bottom) an example of the curvature of the graph of CyclinB1 (purple) and Cell-APP state predictions (black). For Cell-APP, a value of one corresponds to a mitotic prediction, and zero corresponds to a non-mitotic prediction. Graphs displayed are normalized, and the two most prominent local maxima of CylcinB1's curvature are plotted as red dots.

c) Histogram detailing the distribution of the difference between Cyclin B1-inferred and Cell-APP-predicted mitotic entry and exit times (Δt).

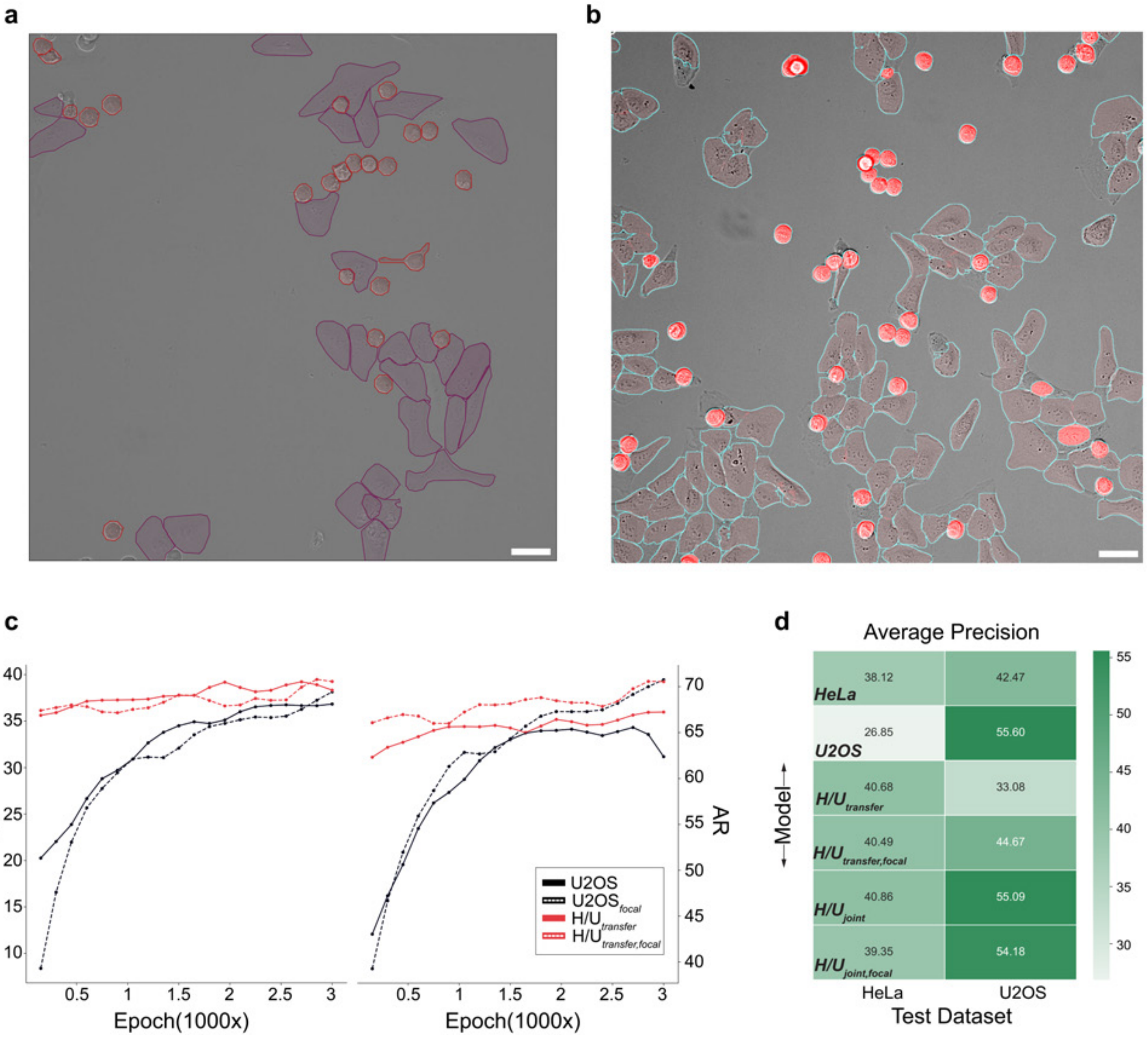

Figure 4: Generalization of Cell-APP annotation and segmentation pipeline and out-of-distribution performance of the trained ViT models

a) Example U2OS annotations generated via Cell-APP methodology (SAM mask prediction and UMAP/HDBSCAN classification). Purple denotes interphase annotation, whereas red denotes m-phase annotation. The sample was illuminated using a brightfield regime and imaged with a 20x objective. Scale bars are roughly 60 microns in length.

b) Cell-APP U2OS dataset trained detector inference example. Inference was conducted with a ViT-large backboned detector trained using focal-loss. Bright red denotes an m-phase prediction, while dim red denotes an interface prediction. Cell borders are given in cyan. The sample was illuminated using a brightfield regime and imaged with a 20x objective. Scale bars are roughly 60 microns in length.

c) Average precision (AP) and average recall (AR) of models training on the Cell-APP U2OS dataset vs. time (training epoch). Statistics were gathered on the held-back testing dataset. Black and purple traces correspond to models trained from ImageNet pre-training checkpoints, while red and beige traces correspond to models trained from Cell-APP HeLa dataset pre-training checkpoints.

d) Average precision heatmap comparing model performances across the HeLa and U2OS testing datasets. All models represented are ViT-large backboned. Model parameters for this experiment were exactly consistent aside from the training dataset and use of focal loss, making the variable

"Model" a surrogate for analyzing the effects of the training dataset and class loss function on performance. The subscript "focal" denotes the use of focal loss, and its absence denotes the use of standard cross-entropy loss. H/U denotes models trained on both the HeLa and U2OS datasets, and the subscript "transfer" or "joint" denotes transfer learning or joint training. In the case of transfer learning, models were first trained on the HeLa dataset and then trained on the U2OS dataset.

Backbone	Dataset	Class Loss	mAP	mAR	$\text{mAP}^{\text{m-phase}}$	$\text{mAR}^{\text{m-phase}}$	Batch Size
ResNet50	HeLa(+)	CE	55.11	72.68	49.66	71.75	4
ViTL	HeLa	FL	57.61	73.01	54.46	72.45	2
ViTB	HeLa(+)	FL	59.95	75.12	57.10	76.53	4
ViTL	HeLa(+)	CE	60.03	75.09	57.43	75.73	2
ViTL	HeLa	CE	60.29	74.153	**58.78**	75.19	2
ViTL	HeLa(+)	FL	**60.55**	**79.28**	57.92	**81.24**	2

Table 1: mAP and mAR scores for detectors trained on the HeLa dataset. Reported scores are measured from the HeLa testing dataset, which did not vary between the HeLa(+) and HeLa sub-datasets. CE = Cross Entropy, FL = Focal Loss, Superscript box and mask refer to the network prediction for which mAP or mAR was measured (either predicted bounding boxes or masks).