Methods in Support of Deep Learning Applications for Three-Dimensional Image Analysis

Nicholas Stitt

Mentor: Dr. Timothee Lionnet

June 30th, 2022

A thesis in fulfillment of the Masters in Biomedical Informatics Degree

SUMMARY

A key challenge in the field of biomedical imaging is the accurate labeling of cell boundaries, known as cell segmentation. Recent breakthroughs in deep learning have led to their application in the automation of cell segmentation. The accurate delineation of cellular boundaries is central to understanding cellular identities and pathological classifications. Developments in microscopy techniques have led to the increased use of three-dimensional imaging to capture the morphological features of cells with greater spatial resolution. These techniques necessitate new methods for annotation and analysis to utilize the increased size of the data generated. While breakthroughs in deep learning segmentation continue, these advancements rarely make it into the hands of the researchers that would benefit from them most. To aid in the effort of deploying these models we have developed two methods which minimize the barrier to entry for utilizing deep learning segmentation. Our method for interpolating layers within three-dimensional images decreases the time and effort needed to perform manual annotation and can generate novel images to extend the utility of limited training sets. We have also developed a function for reliably tracking cells through three-dimensional images, allowing for more precise cellular reconstructions. These methods facilitate the use of deep learning segmentation for researchers who currently lack the means to employ them. As cloud-based computing and data storage become more accessible, increasingly powerful deep learning models can be scaled for use on expanding datasets. This demands standardized methods of capturing and analyzing imaging data to suit a diverse range of cell types and imaging modalities. The methods we have developed support the pursuit of these goals.

INTRODUCTION

Immunofluorescence Microscopy

Microscopy is among the most important and widely used methods for understanding the biological processes at the cellular level (Huang et al., 2009). Understanding the structural and functional relationships between cells remains at the forefront of disease diagnosis and translational research (Lin et al., 2018). Microscopy imaging is an essential technique in scientific and medical research, providing knowledge of tissue organization, down to cellular and subcellular domains (Weigert et al., 2018). Among imaging techniques, immunofluorescence remains one of the most widely used, as it allows for the visualization of structures and specific cellular components through molecule-specific labeling (Huang et al., 2009).

Immunofluorescence imaging utilizes fluorescent probes to tag a targeted protein or molecule of interest. When the fluorophores are exposed to a specific wavelength, they release another wavelength at a specific frequency that can be picked up by a microscope (Lin et al., 2018). The specificity of the tags and the sensitivity of fluorescence microscopes has enabled immense progress in multiple avenues of research, including cancer research (Asante et al., 2021). Beyond nuclei, fluorescent labels can importantly identify the presence of certain proteins, allowing researchers to identify subsets of a cellular population, or use validated biomarkers to classify tumors into molecular subtypes.

Immunofluorescence microscopy has been continually applied and iterated on over the years to cover an expanding list of tasks. Counting individual cells and grouping them based on morphology are two regularly used methods in basic discovery as well as pathology.

Cell Segmentation

The goal of any imaging technique is to generate reliable images for downstream analysis, regardless of the method of acquisition. Because cells are the units of the tissue, a key challenge in the analysis of tissue imaging data is identifying every cell within a tissue. The current explosion in single-cell genomics approaches has made it obvious that tissues represent highly complex mixtures of cell types and states, and that population level summary statistics at best offer limited insight, at worst can miss key phenotypes of cell players (Kashima et al., 2020). The quantity and the morphology of cells are two important factors in identifying disease. These readouts are regularly used in cancer diagnostics which makes accurately identifying these qualities important. Fluorescent imaging can provide a clear indication of the location of targets, making it useful for identification tasks.

Cell segmentation is the task of splitting a microscopic image domain into discrete regions which represent individual instances of cells (Lux and Matula, 2020). The result of this process is a pixel-wise annotation over the identified cells within the image, known as a segmentation mask. It enables downstream analysis of the properties of each cell object, ranging from geometric features, like area, volume, and circularity, to intensity metrics like the brightness of a given marker, homogeneity of signal across the cell, and nuclear localization. These image features represent rich information about the identity and state of each cell, which can be fed into predictive models of disease progression.

Cell segmentation has been highly adopted in the biomedical environments due to the high volume of imaging data that is produced in these fields. But due to the diversity and difference of cell types, there is not a method that can be applied to identify all cell types (Wang and Wang, 2020). To find individual instances of cells or determine the volume of a cell, either the cells' boundaries or nuclei must be annotated on the image (Arbelle et al., 2018). Until recently this process has been done manually. Manually counting cells has multiple drawbacks: the time required, inconsistency between researchers, and the level of expertise required to correctly identify some populations (Cruz-Roa et al., 2014).

Three-Dimensional Imaging in Biomedical Research

Cellular morphology is an indicator of the physiological state of a cell (Lux and Matula, 2020). Most pathology imaging is done on thin tissue sections, which means that only a two-dimensional section of each cell is captured, potentially missing key features that would have been apparent upon imaging the entire volume of the cell and its surrounding microenvironment. Three-dimensional imaging therefore has a unique potential to increase the accuracy of cell classification and understanding of the spatial organization of specific cell populations like tumors. When imaging cells, a common method for collecting three-dimensional data is by capturing z-stacks. A z-stack consists of a set of two-dimensional images acquired one on top of the other through a method known as optical sectioning to computationally reconstruct a three-dimensional volume. Z-stacks are often used with high magnification imaging which limits the amount of background and blur captured in a single plane (Jaques and Liebling, 2020). Multiple forms of fluorescent imaging like confocal microscopy are used to obtain the most reliable images for optical sectioning by filtering out light outside of the focal plane.

The increased spatial information achieved by three-dimensional imaging necessitates stringent cell segmentation. The neurite outgrowth of a neuron is a good example of a task which requires reliable boundaries of a cell that are consistent through multiple layers of an image (Li and Horn, 2006). The workload of manual annotation escalates with increasing sample depth and must be performed in a slice-by-slice manner (Çiçek et al., 2016).

Deep Learning for Image Segmentation

Significant progress in machine learning has allowed us to offload the task of cell segmentation and identification to automated algorithms that can be executed computationally. These algorithms can take an input and transform the data into a usable model for downstream analysis (Zhang et al., 2021). Machine learning systems have been developed to transcribe speech to text, find relevant search results, identify objects in images, and other tasks previously limited to humans (LeCun et al., 2015).

The datasets handled by machine learning approaches can be gigantic: images are data structures with millions of pixels and may be analyzed hundreds at a time. For a machine learning model to work with data usefully, each data instance needs to be summarized into a smaller set of attributes that defines that instance - these attributes are known as features (Zhang et al., 2021). These features are handled within the system as a representation of the input data. The model updates its representations by changing the weights of the features to generate a representation that best matches the desired output. In machine learning, the most common way of training a system is to feed it data that has already been labeled with the desired outcome, known as training data (LeCun et al., 2015).

In practice, this process of creating and modifying representations is done in a processing layer of the model. By stringing together multiple layers, we can pass the representation generated from one layer as the input for the next. When a network consists of multiple layers of these representation-learning methods the network is referred to as deep and the use of these deep networks for machine learning tasks is called deep learning (Schmidhuber, 2015).

Deep learning architectures have been applied with great success to object detection within images (LeCun et al., 2015). Convolutional neural networks are a family of deep networks that were particularly designed for use on two-dimensional data, such as images and videos (Ciregan et al., 2012). The convolution neural network leverages the knowledge that nearby pixels are typically related to each other, allowing them to learn the intensity and composition rules that represent objects within images (LeCun et al., 2015; Zhang et al., 2021) ResNets are another architecture that have seen wide acceptance in recent years due to their ability to train deep networks, like those in image analysis, with greater ease (He et al., 2016).

In recent years, these architectures have led to significant progress in biomedical image analysis (Wang et al., 2019). In some cases, deep learning algorithms even outperform manual annotation in medical image segmentation (Renard et al., 2020). Deep learning methods for quantifying biomedical image data are increasingly implemented to increase throughput, maintain consistency, and reduce human effort (Yang et al., 2020).

Key Issues

The wide applications of deep learning have led to an eruption of research and model development in recent years, but key issues persist (Gu et al., 2018). New algorithms often fail to translate to the labs which may benefit most from them. The correct application of these new models typically requires a large degree of expertise to deploy, hampering progress in the field. (Sugawara et al., 2022).

On top of this, different algorithms may not apply to certain datasets and the workflows that are used (Siriapisith et al., 2021). The characteristics and variety of cells themselves create challenges due to variability in cell shape, cell size, membrane characteristics, cell crowding, and the fact that cell boundaries contain few pixels (Lux and Matula, 2020; Mandal et al., 2021; Wang et al., 2019; Yang et al., 2020).

The workflow for a specific analysis may lie outside the scope of a model, rendering the method unusable without creating workarounds or spending time tuning parameters (Wen et al., 2021). A key example of this is the lack of a reliable algorithm for three-dimensional images (Cao et al., 2020). There have been advances in three-dimensional segmentation, but the field is still small and suffers from the same issues mentioned above (Singh et al., 2020).

Another major impediment to deep learning segmentation is the availability of human annotated datasets (Englbrecht et al., 2021; Greenwald et al., 2021; Ronneberger et al., 2015; Wang et al., 2019)**.** Even with automated cell segmentation techniques accelerating in development, manual expert annotation remains the gold standard to achieve the highest quality of annotation (Falk et al., 2019). Most deep learning algorithms require substantial amounts of annotated training data to generate a robust model (Çiçek et al., 2016). Multi-dimensional datasets containing high levels of spatially accurate data require precise annotation that can be extremely cumbersome to do by hand. As image acquisition instruments increase the amount of data they can generate, the demand for automatic cell annotation is rapidly growing (Falk et al., 2019; Lux and Matula, 2020).

The time required for annotation, the difference in desired outputs, and the variation between datasets leads to a lack of transferability ((Falk et al., 2019; Wang et al., 2019). This problem is exacerbated by an unmet necessity for standardized, easy-to-use software packages (Falk et al., 2019). A solution must be reliable enough to not require image-specific adjustments and it must be accessible to researchers who are unable to set up and evaluate different model architectures (Greenwald et al., 2021; Wang et al., 2019).

While these issues are formidable, attempts are under way to address them. Multiple groups have compiled datasets that are focused on providing sufficient sample objects for cell segmentation while still providing an adequate degree of heterogeneity (Edlund et al., 2021; Schwendy et al., 2020). The last few years have also seen substantial progress on creating an accessible method for researchers to utilize the latest cell segmentation techniques.

An elegant answer to some of these problems is the work of Greenwald et al. (Greenwald et al., 2021). They have developed an open-source approach to annotation that has generated a versatile dataset composed of 1 million manually labeled cells called TissueNet. By incorporating crowdsourcing and a human-in-the-loop approach, they have created a standardized pipeline for annotating a large breadth of cell types, all accessible through a simple web interface. TissueNet is just one part of this group's effort to provide a unified approach for the bottlenecks in cell segmentation today. The group leveraged TissueNet to train their state-of-the-art deep learning pipeline for cell segmentation called Mesmer. Mesmer's deep learning model is composed of feature pyramid networks, an architecture which excels in recognizing objects at different scales (Lin et al., 2016). Each of these feature pyramid networks are structured around a ResNet, which predicts the classification of each pixel in the image. The Mesmer pipeline is more than just a deep learning algorithm. It incorporates preprocessing of images by performing normalization techniques to handle variations in the image and it performs post-processing that transforms the continuous predictions into discrete labels. Importantly, access to the Mesmer pipeline and the images in TissueNet are available online through a web interface or it can be downloaded as a package to be run locally.

Our Contributions

Efforts like Mesmer are a step forward in bringing these algorithms to the biomedical research community, but it retains some of the limitations present within the field as a whole. While pipelines like

TissueNet ease the task of manual annotation, it remains a time-consuming challenge. Manual annotation of segmentation masks will always need to be performed regardless of the quality of segmentation algorithms. Even with hand annotated masks, variability within the dataset often means that a certain cell or tissue type will not be applicable to an off the shelf method (Greenwald et al., 2021). Many biomedical research labs do not produce images in the quantity necessary to generate rigorous training data which can lead to poor model performance (Çiçek et al., 2016). The limitations of widely available pipelines also apply to three-dimensional datasets (Cao et al., 2020). Segmentation algorithms, including Mesmer, can struggle to track cells in three-dimensions, meaning key information on cell characteristics may be lost due to poor spatial tracking. Our work intends to help fill these gaps that exist in the current annotation and segmentation pipelines.

The dataset we used consists of 23 images collected from ovarian cancer tissue samples at the NYU Grossman School of Medicine. Ovarian cancer ranks among the most lethal malignancies for women and its early detection is key for increasing the survival rate in patients (Jacobs and Menon, 2004; Yang et al., 2021). Malignant cells often display morphological abnormalities, which can make accurate segmentation with automated methods challenging (Fischer, 2020). The quantity and the morphology of cells are two important factors in identifying disease and are readouts regularly used in cancer diagnostics which makes accurately identifying these qualities crucial. For instance, mitotic nuclei are routinely identified as an indicator of the degree of malignancy in human breast cancer histopathology samples (Yang et al., 2020).

Each of these samples were labeled with the nuclear marker DAPI and imaged using a confocal microscope as a z-stack. Each z-stack is composed of seven sequential images. Cell segmentation masks have been manually generated for every plane of these images. This dataset provides us a strong basis for exploring the range of the state-of-the-art segmentation techniques while ensuring that it can be applied to a critical need.

First, we have developed a z-stack Interpolation pipeline to accelerate the annotation of z-stacks. This method infers the segmentation result of an intermediate layer of the z-stack based on the annotations of the layers under and above it, using the fact that nuclei are generally continuous objects with simple shapes. The automated generation of intermediate masks reduces the amount of time needed to manually annotate a three-dimensional dataset: manual annotation can be done on a small number of layers subsampling a z-stack vertically, followed by z-stack interpolation to generate annotations in the intervening layers. The interpolated masks can be updated by the researcher to address any areas where the mask does not align with the associated image, a process that is much less time consuming.

The second use case of z-stack interpolation capitalizes on the function’s interpolation abilities to generate completely novel images. Using the interpolated segmentation masks in conjunction with the raw images, we can generate new images based on the properties of the surrounding layers. This method can in principle automatically extend the size of a training dataset beyond standard data augmentations.

The second method we have developed supports segmentation analysis or cell tracking in 2D over time, or in three dimensions. Our approach relies on accurately tracking the segmentation masks throughout a z-stack to preserve three-dimensional properties of the cells. Our function is called CellTracker and generates reliable cell tracking across z-stacks when compared to methods like Mesmer. CellTracker allows for more precise reconstruction of three-dimensional data and allows for a broader range of imaging pipelines to be applied to available segmentation techniques.

METHODS

Z Stack Interpolation on Masks

Interpolated layers are generated by specifying a number of intermediate layers to create in between two input masks. The two input mask given to the function are known as the boundary layers. For the interpolation of masks, the pixels of the boundary layers are binary, indicating if the pixel belongs to a cell or the background within the input image. The binary masks of an image containing multiple layers are passed to the function along with the desired number of intermediate images to interpolate (Figure 1A).

The first step of the process takes the union and the intersection of the positive pixels in the two boundary layers. The difference between the union and the intersection is found and used as the region of interest for interpolation. This region is used because it is where the boundary masks have different annotations (Figure 1B). The distance map for each boundary mask is generated using the distance_transform_cdt function from the scipy package. The distance map is simply the Euclidean distance from each positive pixel to the nearest background pixel. A pixel with a high value indicates that it is close to the center of the cell it belongs to.

The boundary layers are assigned as the start and end planes according to their position in the initial z-stack. The number of intermediate layers being interpolated is used to determine what the distance will be between the boundary layers in the final output stack. In the case of interpolating three new masks, the start boundary layer would be referred to as layer one, the end boundary layer would be referred to as layer five, and the difference between the boundary layers would be four.

Each intermediate mask is derived from a combination of the boundary masks. The only pixels that must be interpolated are those in the disputed region; consensus pixels are carried over. This decision is based on the relative distance of the intermediate plane with each boundary layer using a weight system.

The pixels taken from each boundary image are found by first determining the weight of each boundary layer in relation to the intermediate layer within the new stack (Figure 1C). The weight of the starting boundary mask for each intermediate layer being generated is given by

$$\mathrm{WZ}_{\mathrm{Start}} = ((\mathrm{Diff}+1) - \mathrm{Z}_{\mathrm{New}}) \text{ x } 1/\mathrm{Diff},$$

where $\mathrm{WZStart}$ is the weight for the starting boundary mask, Diff is the difference between the boundary layers in the final stack, and $\mathrm{Z}_{\mathrm{New}}$ is the position of the mask being interpolated within the new stack. The weight of the ending boundary mask for each intermediate layer being generated is given by

$$\mathrm{WZ}_{\mathrm{End}} = 1\text{-}\mathrm{WZ}_{\mathrm{Start}},$$

where $\mathrm{WZ}_{\mathrm{End}}$ is the weight for the ending boundary mask.

These weights are then applied to the distance maps of each of the boundary masks. For each boundary mask, the maximum distance within each cell is found and multiplied by the weight for that boundary layer, producing a cutoff score. Pixels with a distance value greater than that cutoff score are carried over to the intermediate mask (Figure 1D).

The final mask for the intermediate layer is found by taking the intersection of the inherited pixels with the disputed region. This image is then combined with the intersection of the boundary images to produce the final image (Figure 1E).

Z Stack Interpolation on Images

The interpolation of images is carried out after masks have been interpolated using the above method. The intermediate mask that is generated is overlaid with input boundary images. The weighted average of the two images are found using the “addWeighted” function from the OpenCV2 package. The weights used for this function are the same as those used in the mask interpolation step.

CellTracker

A z-stack of masks is passed through the CellTracker function to accurately track the cells throughout the stack and preserve the three-dimensional properties of the cells. The scikit-learn package is used to label the masks, which results in individual cells being labeled starting from one on each layer (Figure 2B). The input masks, the labeled masks, and the region properties for each z-layer are stored in a dictionary representing the original organization of the labels. The area, x-position of the centroid, and y-position of the centroid of each label are collected as the region properties for each cell and placed into a dataframe. A copy of the original dictionary is created to store the updated labels.

The first step is to move through each stack in the forward direction, from the top layer to the bottom, and rename each label to a unique label within the stack (Figure 2C). The labels that have an area smaller than 50 pixels^2 are removed.

To accurately label the stack, we start by moving through each layer in a single direction. The cell morphology for each cell between two layers must be classified for accurate labeling. The six classifications for cell morphology between layers are static, growing, shrinking, started, ended, branching, and joining (Figure 2A). The classifications are inherited in the forward direction, with the cells in the bottom layer associating with the cells in the bottom layer. The initial association is done using the overlap of cells between the top and bottom layers. A cell from the top layer that overlaps a single cell in the bottom layer is classified as static, growing, or shrinking depending on the relationship between the cell sizes. Cells in the bottom layer inherit the labels from the top layer in this scenario. A cell in the bottom layer that does not overlap with any cells in the top layer indicates a cell that is starting. No labels are inherited in this scenario. A cell from the top layer that does not overlap with any cells in the bottom layer indicates the cell has ended and no labels are inherited in this scenario, either.

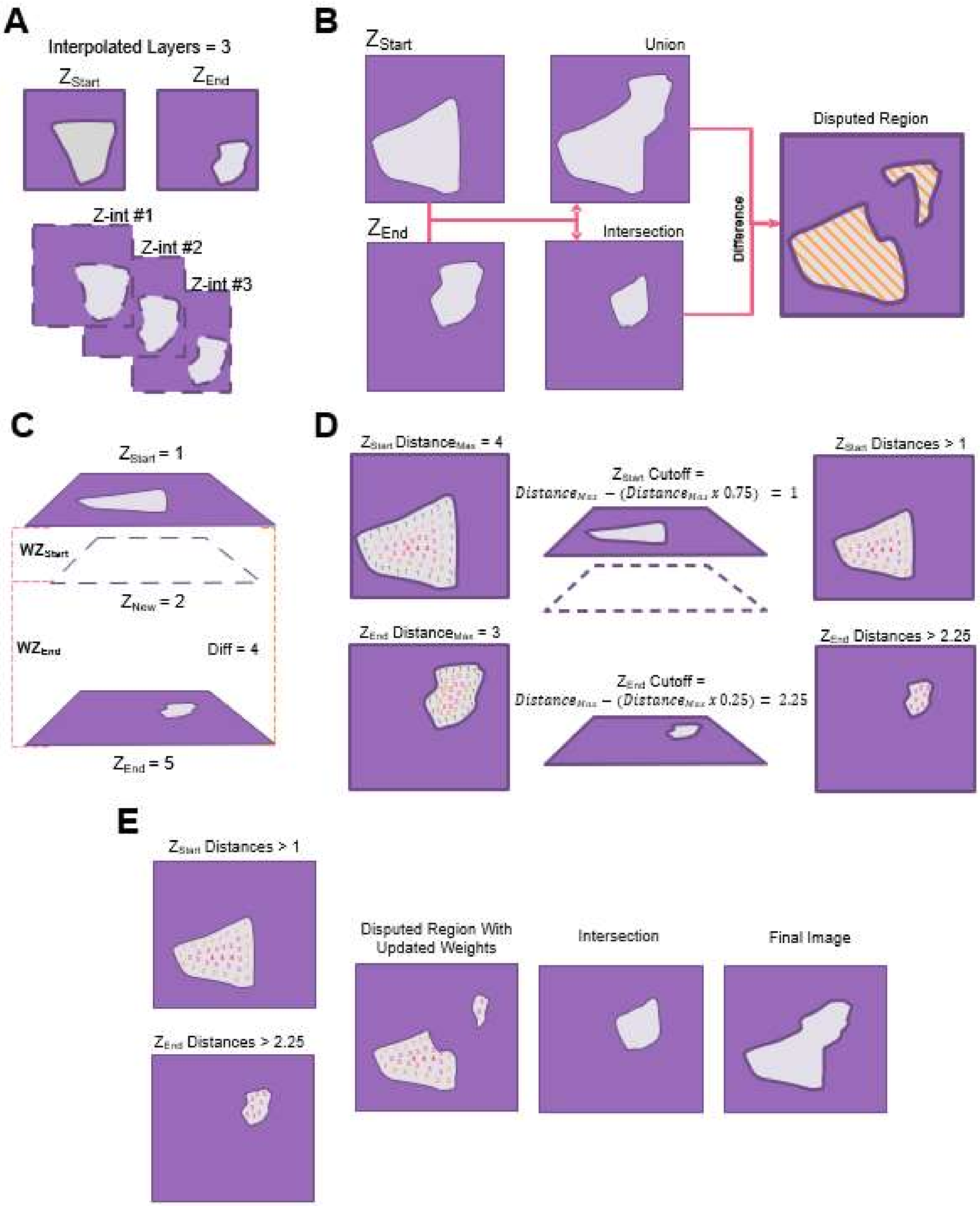
A
Interpolated Layers = 3
Z_{Start}
Z_{End}
Z-int #1
Z-int #2
Z-int #3
B
Z_{Start}
Union
Disputed Region
Z_{End}
Intersection
Difference
C
$Z_{Start} = 1$
WZ_{Start}
$Z_{New} = 2$
WZ_{End}
Diff = 4
$Z_{End} = 5$
D
Z_{Start} $Distance_{Max} = 4$
Z_{Start} Cutoff =
$Distance_{Max} - (Distance_{Max}\ x\ 0.75) = 1$
Z_{Start} Distances > 1
Z_{End} $Distance_{Max} = 3$
Z_{End} Cutoff =
$Distance_{Max} - (Distance_{Max}\ x\ 0.25) = 2.25$
Z_{End} Distances > 2.25
E
Z_{Start} Distances > 1
Z_{End} Distances > 2.25
Disputed Region With Updated Weights
Intersection
Final Image

Figure 1. Workflow for the Interpolation of Intermediate Images Within a Z-Stack

(A) Example of two boundary masks with a desired interpolation of three intermediate masks.

(B) The intersection and union of the boundary masks are used to identify the areas with disputed annotations.

(C) The method of finding the input weights of the boundary masks, shown using the top of the three masks to be interpolated.

(D) The weights of the boundary images are applied to their distance maps to produce modified distance maps.

(E) The modified distance maps are inherited along with the initial intersection to produce the final image

A common situation is for multiple labels from one layer to be associated with a single label from the other layer. When this happens in the forward direction it is classified as cell branching. When this happens in the reverse direction it is classified as cell joining. Both situations describe the same arborized cell, but the direction matters for how we assess the labelling. This feature of label overlap also occurs due to the natural variation of cell sizes in a tightly packed environment, meaning that it cannot be assumed that every overlap is due to an arborized cell.

These scenarios require an extra level of assessment to determine the correct method for labeling. This is achieved by looping through each cell of the bottom layer that has an overlap with a label in the top layer. The percent of the cell in the bottom layer that overlaps with the cell in the top layer is calculated. If the overlapping cell area is greater than 50% of the cell's total area, it is assumed that the cell is a continuation of the cell in the top layer. In this situation, the cell in the bottom layer inherits the label from the cell in the top layer (Figure 2D).

This method can only be used to change the label for cells that are branching by updating the labels in the bottom layer. In the scenario where cells are joining, changing the labels in the top layer is not viable. The correct cell label would not be propagated forward through the stack in proceeding iterations. These labels are left unchanged during the forward propagation.

The final step for identifying cells that are joining is to do a reverse iteration through the z-stack focused only on the cells that were left unchanged in the first forward pass. These labels are treated the same way as the branching cells in the forward pass to update the labels correctly (Figure 2E).

The final output of this function is the updated dictionary which contains the masks, new labels, and region properties table for each layer in the z-stack. We have also developed a suite of functions for displaying the stack with the uncolored labels, the labels with their corresponding label number overlaid on each cell, and the colored labels using the Glasbey palette (Glasbey et al., 2007). These functions also allow for the mask and label z-stacks to be extracted from the output dictionary and saved as either a tiff or gif file for visualization.

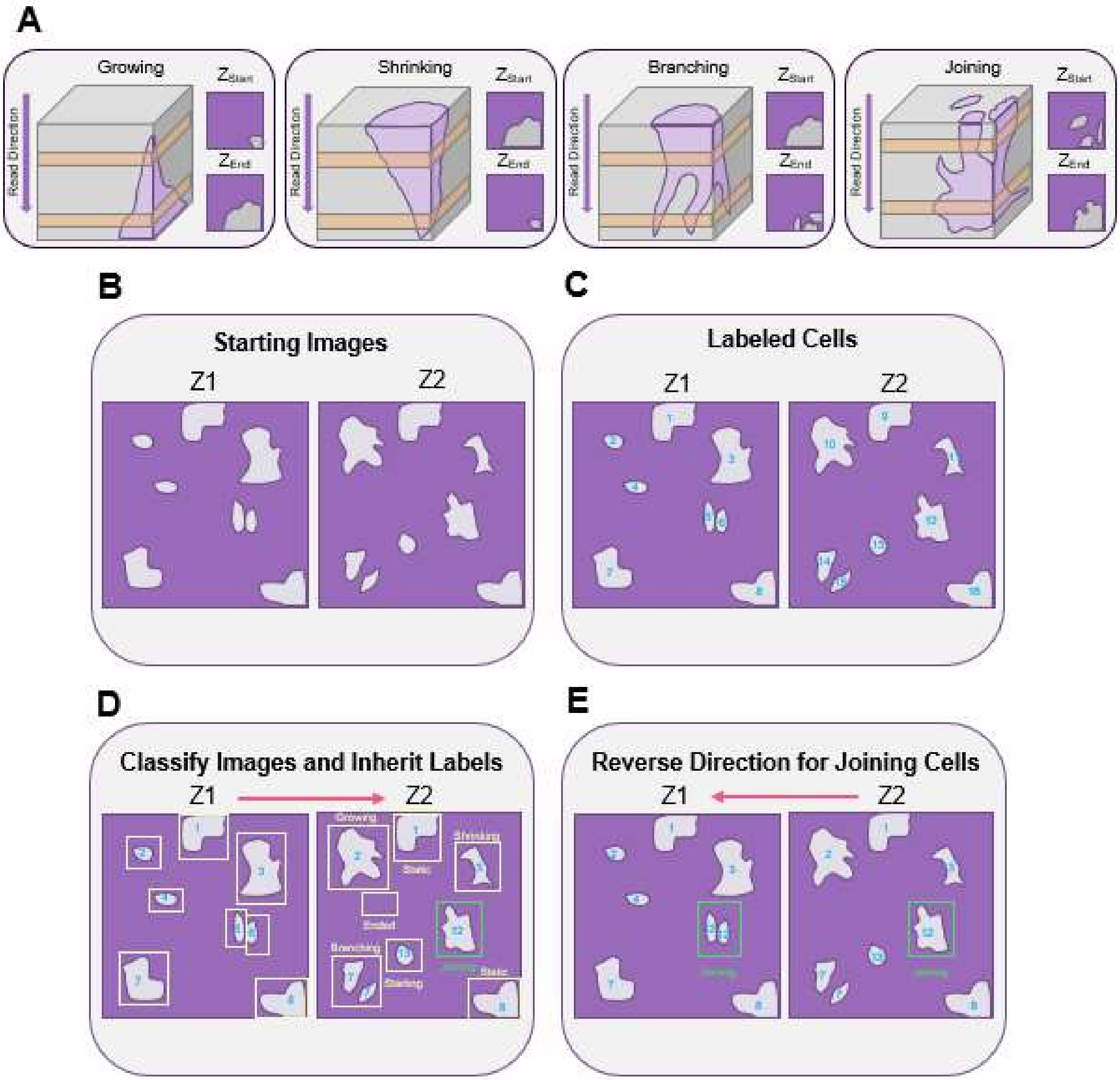

Figure 2. Workflow for the CellTracker Method for Labeling Cells Through Three-Dimensional Images

(A) Representations of four common morphological cell classifications. The left side of each panel shows the representation of the cell morphology in a three-dimensional voxel, with two orange stripes signifying imaging planes. The images generated from each of those planes are shown on the right.

(B) Two sequential masks from a z-stack that are unlabeled.

(C) The initial labels assigned to the cells, which are unique between both planes.

(D) The propagation of all labels in the forward direction except for cells which are joining. The bounding boxes from Z1 are overlaid on Z2 and labeled to indicate the morphological classification between the layers. All cells in Z2 have inherited the corresponding labels from Z1 except for cells which are joining, which are left unchanged (highlighted in green).

(E) Label propagation occurs in the reverse direction to update the cells which were classified as joining.

RESULTS

Z Stack Interpolation on Masks

The manually annotated masks were used as ground-truth annotations to quantify the successful interpolation of intermediate masks. We interpolated masks between layers where a manually annotated mask already existed. This allowed us to compare the generated masks to their ground-truth counterparts (Figure 3A).

This process was used to test the quality of interpolation using different interpolation sizes. Interpolation of one, two, and three masks was tested within each of the 23 masks of the dataset. While each original z-stack is composed of seven images, the z-stack interpolation generates z-stacks composed of five images. This discrepancy is caused by the lack of boundary masks surrounding the start and end masks, leading to an inability to interpolate the intermediate mask. The start and end masks were removed from the input masks for comparison with the interpolated masks to account for this.

The mean square error (MSE) was used as the metric of accuracy for these interpolations due to the binary nature of the masks. The MSE is a value between zero and one, with a smaller value indicating greater accuracy. The mean MSE for all masks when interpolating one intermediate mask was 3.61 x 10^{-2} (SEM = 1.28 x 10^{-3}). The MSE increased with an interpolation distance of two to 5.51 x 10^{-2} (SEM = 1.84 x 10^{-3}). This trend continued when the interpolation distance was increased to three with a MSE of 7.74 x 10^{-2} (SEM = 2.57 x10^{-3}) (Figure 3B).

The accuracy of the interpolation decreases as the distance is increased, which was expected. It is also observed that the spread of the MSE increases as the distance is increased. This suggests that any errors that are present in the initial interpolation are further exaggerated as the distance increases.

Z Stack Interpolation on Images

Interpolated images were generated with the same dataset by overlaying the interpolated mask and boundary weights on the raw boundary images (Figure 3C**.** The structural similarity index (SSIM) was used as a metric of accuracy for comparing the interpolation of images due to the formula's emphasis on interdependencies between pixels that are spatially close (Wang et al., 2004). The SSIM is a value between zero and one, with a higher value indicating greater accuracy.

The same method of comparing the interpolation of masks was used to compare the interpolation of the images. The mean SSIM for all images when interpolating one intermediate image was 6.77 x 10^{-1} (SEM = 1.08 x 10^{-2}). A slight decrease in the SSIM was observed when interpolating two intermediate images with a mean SSIM of 6.59 x 10^{-1} (SEM = 1.14 x 10^{-2}). Interpolating three intermediate images produced a mean SSIM of 6.46 x 10^{-1} (SEM = 1.17 x 10^{-2}) (Figure 3D).

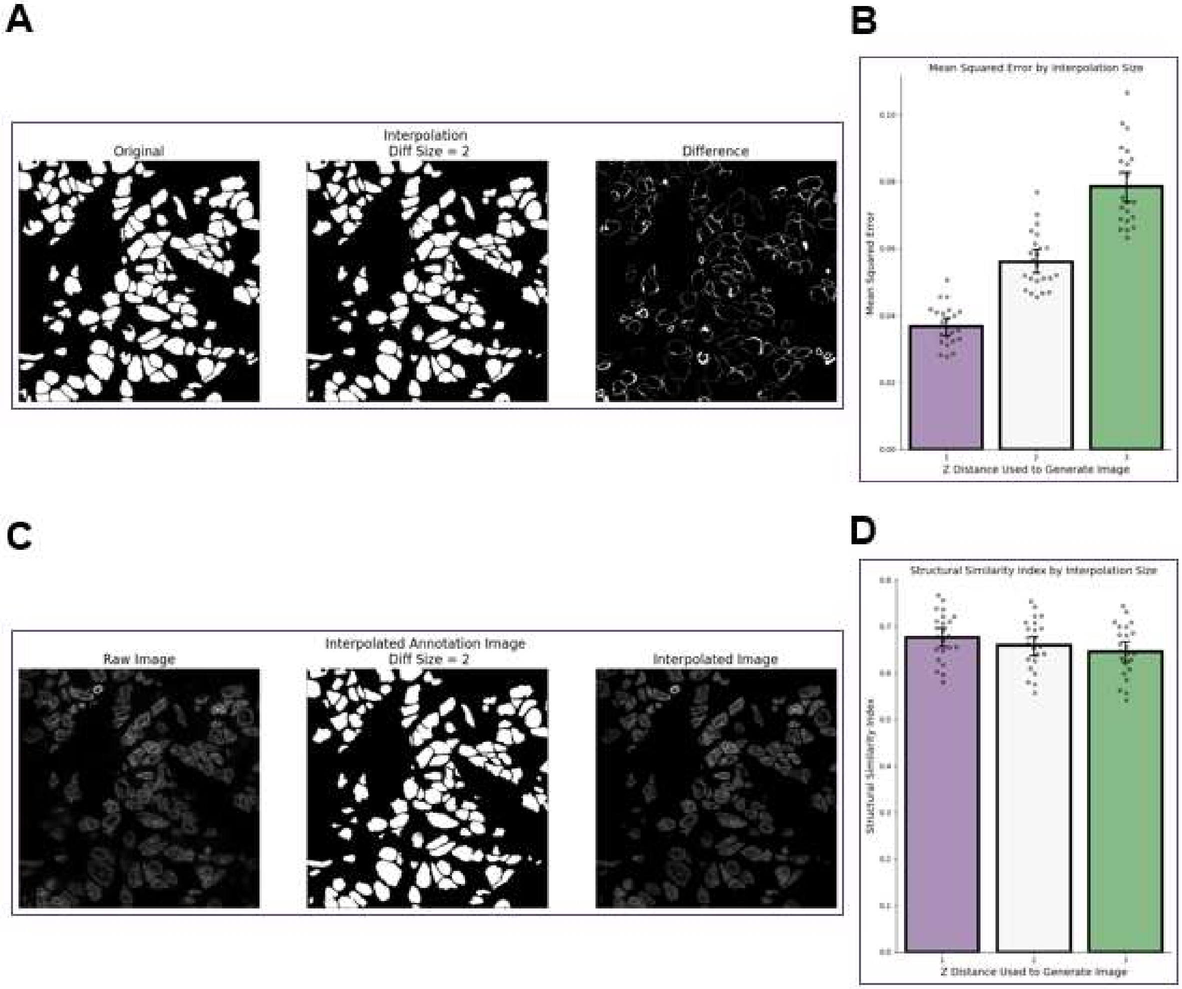

Figure 3. The Results of Z-Interpolation for Mask and Image Generation

(A) Comparison of the ground-truth annotation with the interpolated mask at the same layer. The difference found by subtracting one image from the other highlights pixels that were missed in interpolation.

(B) The mean squared error between interpolated masks and their ground-truth counterparts. Distances between boundary layers of one, two, and three layers were tested, showing a decrease in accuracy with greater distance.

(C) Example of image interpolation. The ground-truth image is shown next to the interpolated mask used to generate the final interpolated image.

(D) The results of image interpolation shows a modest decrease in accuracy as the distance size increases.

Increasing the interpolation distance for images lead to a decrease in interpolation accuracy and an increase in spread, though not to the extent that was observed in the mask interpolation.

CellTracker Validation

Many of the automated image segmentation techniques today offer accurate cell segmentation within a given image but tracking cells through three-dimensions remains limited. Community-focused solutions like Mesmer offer functions for tracking cells but produce unreliable results. We have created a function called CellTracker which can more reliably track cells using the associated segmentation masks.

The manually annotated segmentation masks of our dataset were used to validate the efficacy of CellTracker. The cell numbers and volumes for the dataset were not available, meaning the methods could not be compared against ground-truth values. We used internal validation alongside visualization to confirm accuracy of the CellTracker method. Internal validation was accomplished by running the function on each z-stack in the forward direction and again in the reverse direction. The number of cells and the volume of cells were compared between the two run directions with a decrease in the difference between the runs being used as a metric for accuracy.

The total cell counts and average cell volumes for each z-stack in the dataset were collected in the forward and reverse direction for comparison. Across the dataset, the average cell count in the forward direction was 167.17 cells (SEM = 6.95) and the average cell count in the reverse direction was 164.78 (SEM = 7.02), giving an difference of 2.39 cells or 1.43% of the forward cell count. The average volume of cells in the forward direction was 12839.01 μm^3 (SEM = 404.47) and the average volume of cells in the reverse direction was 13043.20 μm^3 (SEM = 416.62). This is a difference of 204.19 μm^3 or 1.59% of the forward cell volume (Figures 4A and 4B). These results confirm that CellTracker was consistently labeling cells between directions.

Another reliable method for confirming label consistency was to use a three-dimensional representation of the cell labels. These volumetric representations were created by applying the label color throughout the entirety of the z-stack. This method highlights any discrepancies in cell labeling between layers and confirms the uniformity of the labeling in both directions (Figure 4C).

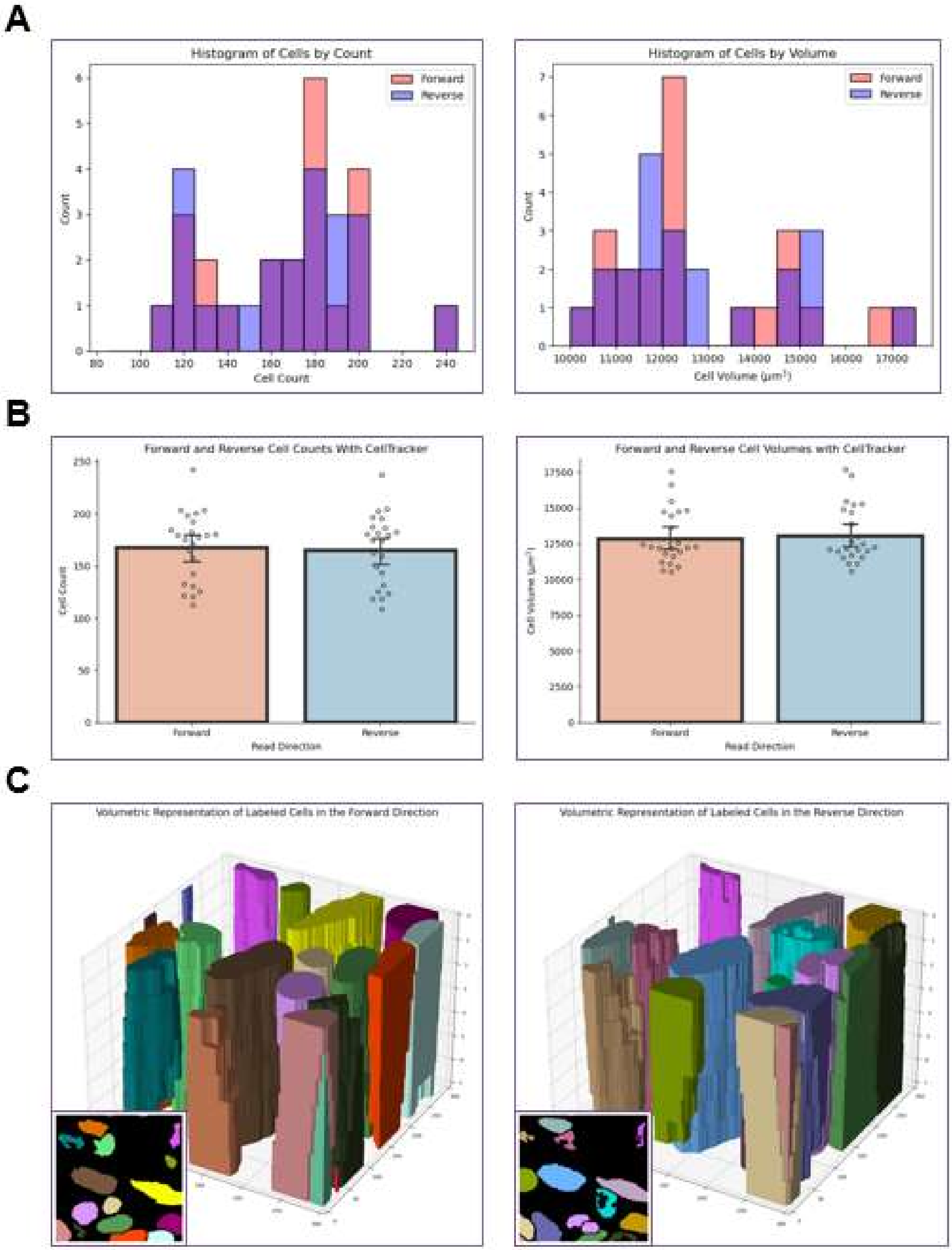

Figure 4. Internal Validation of the CellTracker Method

(A) Histograms highlighting the range and overlap of distributions for cell count and volume between the forward and reverse passes with the CellTracker function.

(B) Quantitative results of the cell count and volume showing the consistency between the forward and reverse passes of CellTracker.

(C) Volumetric reconstruction of the cells using the corresponding label color at each layer. Inset image shows the colored labels from each top layer of each z-stack.

CellTracker Comparison with Mesmer

To compare Mesmer's cell tracking with our CellTracker method, we passed the raw image z-stacks from our dataset through Mesmer for single cell segmentation. The resulting segmentation masks were supplied to Mesmer's cell tracking function alongside CellTracker. This was done to maintain consistency when comparing the two methods. Both tracking methods were run in forward and reverse on the z-stacks to test the internal accuracy of the methods.

Mesmer's cell tracking produced an average cell count of 269.04 cells (SEM = 8.00) and an average cell volume of 7869.97 μm^3 (SEM = 202.93). CellTracker produced an average cell count of 165.98 cells (SEM = 6.97) and an average cell volume of 12941.10 μm^3 (SEM =409.30) (Figure 5A). There is a sizable difference between Mesmer and CellTracker, but it is difficult to make any conclusions without ground-truth data.

To investigate the robustness of the methods, we applied the same internal validation method that was used to benchmark CellTracker. Mesmer had an average cell count difference of 5.39 cells (SEM = 8.17 x 10^{-1}) and an average volume difference of 152.96 μm^3 (SEM = 22.03) between the forward and reverse directions. CellTracker achieved an average cell count difference of 3.52 cells (SEM = 6.12 x 10^{-1}) and an average volume difference of 283.46 μm^3 (SEM = 49.52) between the forward and reverse directions (Figure 5B).

CellTracker showed a smaller variability between read directions for cell counts when compared to Mesmer. Conversely, CellTracker had a greater variability in cell volume between read directions when compared to Mesmer. These results do not indicate which method is more reliable on their own, so we employed volumetric projections to verify the tracking. The volumetric projections of the methods show an observable difference in the consistency of cell tracking through the stack, with CellTracker achieving more reliable labels (Figure 5C).

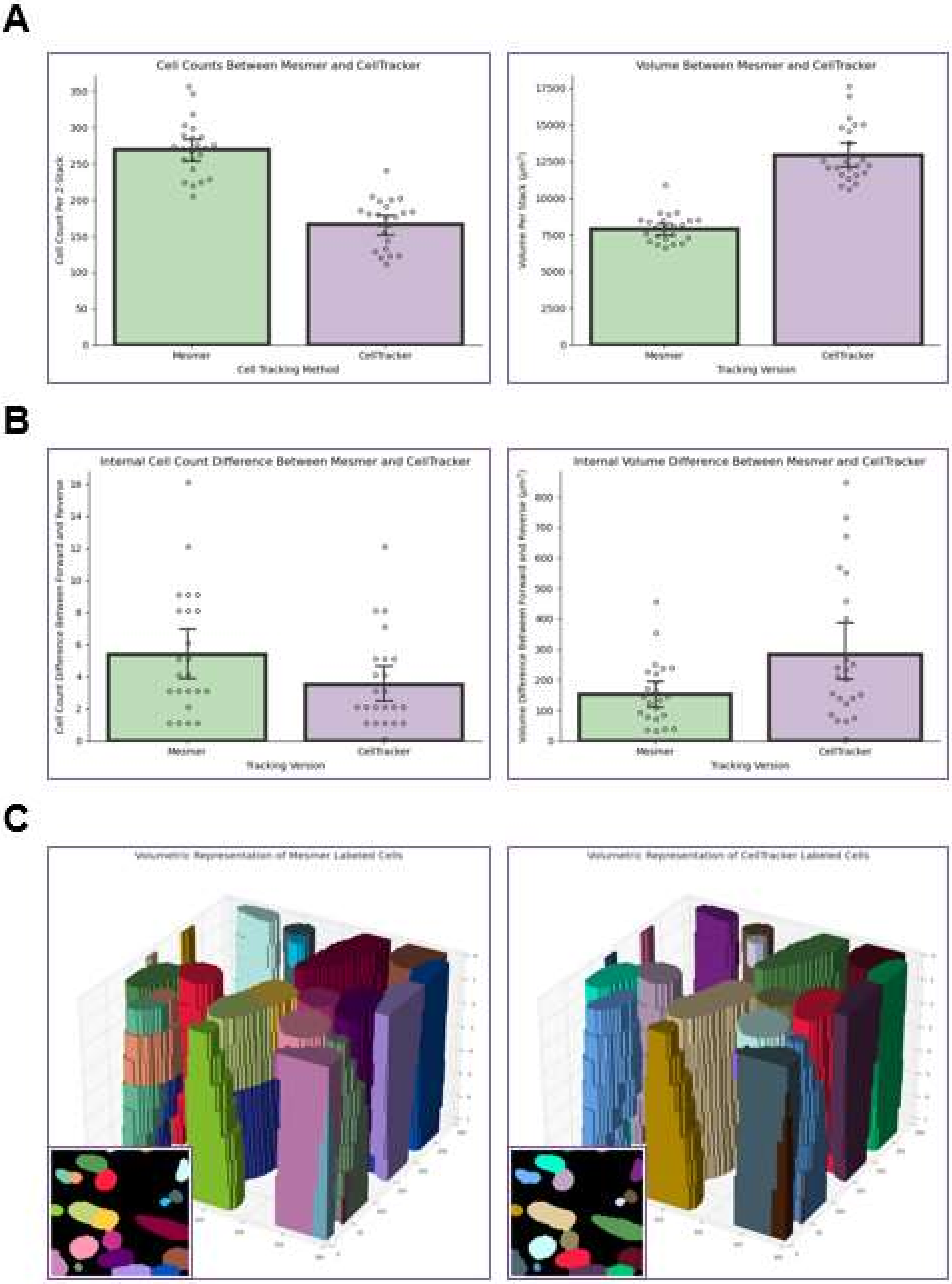

Figure 5. Comparison of Three-Dimensional Cell Tracking Between Mesmer and CellTracker

(A) Quantitations of the total cell counts and volume identified between Mesmer and CellTracker.

(B) Differences between forward and reverse passes with each method for total cell count and average cell volume. Smaller differences between passes indicates greater method consistency.

(C) Volumetric reconstructions using both methods. Reconstruction of Mesmer labeled cells shows a lack of label uniformity through the z-stack when compared to the CellTracker reconstruction.

DISCUSSION

The methods outlined here aim to help researchers and the general scientific community to better apply deep learning techniques with their workflows. We have shown that z-stack interpolation is a reliable method for expediting annotation and can increase the size of training data for deep learning applications. We have also presented a method which will help researchers advance the utility of single cell segmentation into the third dimension.

An aim for this project is to make these algorithms more accessible through an open-source platform or to integrate them into an existing pipeline like Mesmer and TissueNet. Supporting these efforts allows the knowledge base to grow and opens the door for more support from other researchers. With the increased scalability of cloud computing and growing applicability of models, online cell segmentation websites with simple interfaces allows researchers to train a model using their own images (Bannon et al., 2020; Greenwald et al., 2021; Stringer et al., 2020).

We hope to continue development on these projects in the future. A current aim for image interpolation is to identify methods for updating the background of the interpolated images. The current method drops all background data which leads to lower SSIM score and may impede segmentation algorithms. The same algorithms which were applied to produce an averaged image may be applied to the background of these images to create a smoother result.

Another area of focus is to apply CellTracker to datasets which have been annotated in three-dimensions to quantify accuracy. The lack of ground-truth controls to compare the cell labeling poses a challenge for ensuring consistent cell labeling and makes improvements based on quantitative results difficult.

It is not lost on us that these techniques are also fitting for use within other biomedical fields. We hope to be able to explore the application of these methods to other communities with a focus on three-dimensional reconstruction like CT scans and MRIs (Hsu et al., 2021; Siriapisith et al., 2021). For instance, the volume of the epicardial fat surrounding the heart has an important significance to major adverse cardiovascular events. This volume is measured through CT images, but the variability and small size of the fat deposits make its accurate segmentation one of the most difficult tasks in segmentation (Siriapisith et al., 2021). Efficient tracking through the methods we have proposed has the potential to make this volume estimation more practical.

Deep learning image segmentation will continue to grow in popularity and accuracy. As better algorithms are discovered, it is key that they are available for use by the community at large. Open-source platforms and a growing library of methods will prove indispensable for working with tissue imaging data. Approaching these challenges with scalable deployment in mind will ensure that these breakthroughs make it into the labs of researchers who can benefit from them most.

Acknowledgements

The author thanks Dr. Timothee Lionnet for his mentorship and advice through this project; Dr. Kelly Ruggles, Sandra Squarcia, and Dr. David Fenyo for their coordination and support throughout this program; and Dr. Stratos Efstathiadis for enabling this project by providing me sponsorship for the Greene Cluster.

Code Availability

All code used in this project is publicly available at https://github.com/NStitt/Practicum2022.

REFERENCES

Arbelle, A., Reyes, J., Chen, J.-Y., Lahav, G., and Riklin Raviv, T. (2018). A probabilistic approach to joint cell tracking and segmentation in high-throughput microscopy videos. Med. Image Anal. *47*, 140–152. https://doi.org/10.1016/j.media.2018.04.006.

Asante, D.-B., Morici, M., Mohan, G.R.K.A., Acheampong, E., Spencer, I., Lin, W., van Miert, P., Gibson, S., Beasley, A.B., Ziman, M., et al. (2021). Multi-Marker Immunofluorescent Staining and PD-L1 Detection on Circulating Tumour Cells from Ovarian Cancer Patients. Cancers *13*. https://doi.org/10.3390/cancers13246225.

Bannon, D., Moen, E., Schwartz, M., Borba, E., Kudo, T., Greenwald, N., Vijayakumar, V., Chang, B., Pao, E., Osterman, E., et al. (2020). DeepCell Kiosk: Scaling deep learning-enabled cellular image analysis with Kubernetes.

Cao, J., Guan, G., Ho, V.W.S., Wong, M.K., Chan, L.Y., Tang, C., Zhao, Z., and Yan, H. (2020). Establishment of a morphological atlas of the Caenorhabditis elegans embryo using deep-learning-based 4D segmentation. Nat. Commun. *11*. https://doi.org/10.1038/s41467-020-19863-x.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., and Ronneberger, O. (2016). 3D U-net: Learning dense volumetric segmentation from sparse annotation. Lect. Notes Comput. Sci. *9901 LNCS*, 424–432. https://doi.org/10.1007/978-3-319-46723-8_49.

Ciregan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 3642–3649. https://doi.org/10.1109/CVPR.2012.6248110.

Cruz-Roa, A., Basavanhally, A., González, F., Gilmore, H., Feldman, M., Ganesan, S., Shih, N., Tomaszewski, J., and Madabhushi, A. (2014). Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. Medical Imaging 2014: Digital Pathology *9041*, 904103. https://doi.org/10.1117/12.2043872.

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

Englbrecht, F., Ruider, I.E., and Bausch, A.R. (2021). Automatic image annotation for fluorescent cell nuclei segmentation. PLoS One *16*, e0250093. https://doi.org/10.1371/journal.pone.0250093.

Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., et al. (2019). U-Net: deep learning for cell counting, detection, and morphometry. Nat. Methods *16*, 67–70. https://doi.org/10.1038/s41592-018-0261-2.

Fischer, E.G. (2020). Nuclear Morphology and the Biology of Cancer Cells. Acta Cytol. *64*, 511–519. https://doi.org/10.1159/000508780.

Glasbey, C., van der Heijden, G., Toh, V.F.K., and Gray, A. (2007). Colour displays for categorical images. Color Res. Appl. *32*, 304–309. https://doi.org/10.1002/col.20327.

Greenwald, N.F., Miller, G., Moen, E., Kong, A., Kagel, A., Dougherty, T., Fullaway, C.C., McIntosh, B.J., Leow, K.X., Schwartz, M.S., et al. (2021). Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. Nat. Biotechnol. https://doi.org/10.1038/s41587-021-01094-0.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al. (2018). Recent advances in convolutional neural networks. Pattern Recognit. *77*, 354–377. https://doi.org/10.1016/j.patcog.2017.10.013.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition *2016-Decem*, 770–778. https://doi.org/10.1109/CVPR.2016.90.

Hsu, L.-M., Wang, S., Walton, L., Wang, T.-W.W., Lee, S.-H., and Shih, Y.-Y.I. (2021). 3D U-Net Improves Automatic Brain Extraction for Isotropic Rat Brain Magnetic Resonance Imaging Data. Front. Neurosci. *15*, 801008. https://doi.org/10.3389/fnins.2021.801008.

Huang, B., Bates, M., and Zhuang, X. (2009). Super-resolution fluorescence microscopy. Annu. Rev. Biochem. *78*, 993–1016. https://doi.org/10.1146/annurev.biochem.77.061906.092014.

Jacobs, I.J., and Menon, U. (2004). Progress and Challenges in Screening for Early Detection of Ovarian Cancer *. Mol. Cell. Proteomics *3*, 355–366. https://doi.org/10.1074/mcp.R400006-MCP200.

Jaques, C., and Liebling, M. (2020). Aliasing mitigation in optical microscopy of dynamic biological samples by use of temporally modulated color illumination and a standard RGB camera. J. Biomed. Opt. *25*. https://doi.org/10.1117/1.JBO.25.10.106505.

Kashima, Y., Sakamoto, Y., Kaneko, K., Seki, M., Suzuki, Y., and Suzuki, A. (2020). Single-cell sequencing techniques from individual to multiomics analyses. Exp. Mol. Med. *52*, 1419–1427. https://doi.org/10.1038/s12276-020-00499-2.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature *521*, 436–444. https://doi.org/10.1038/nature14539.

Li, C., and Horn, J.P. (2006). Physiological classification of sympathetic neurons in the rat superior cervical ganglion. J. Neurophysiol. *95*, 187–195. https://doi.org/10.1152/jn.00779.2005.

Lin, J.-R., Izar, B., Wang, S., Yapp, C., Mei, S., Shah, P.M., Santagata, S., and Sorger, P.K. (2018). Highly multiplexed immunofluorescence imaging of human tissues and tumors using t-CyCIF and conventional optical microscopes. Elife *7*. https://doi.org/10.7554/eLife.31657.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2016). Feature Pyramid Networks for Object Detection.

Lux, F., and Matula, P. (2020). Cell Segmentation by Combining Marker-Controlled Watershed and Deep Learning.

Mandal, D., Vahadane, A., Sharma, S., and Majumdar, S. (2021). Blur-Robust Nuclei Segmentation for Immunofluorescence Images. Conf. Proc. IEEE Eng. Med. Biol. Soc. *2021*, 3475–3478. https://doi.org/10.1109/EMBC46164.2021.9629787.

Renard, F., Guedria, S., Palma, N.D., and Vuillerme, N. (2020). Variability and reproducibility in deep learning for medical image segmentation. Sci. Rep. *10*, 13724. https://doi.org/10.1038/s41598-020-69920-0.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation.

Schmidhuber, J. (2015). Deep learning in neural networks: an overview. Neural Netw. *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003.

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

Singh, S.P., Wang, L., Gupta, S., Goli, H., Padmanabhan, P., and Gulyás, B. (2020). 3D Deep Learning on Medical Images: A Review. Sensors *20*. https://doi.org/10.3390/s20185097.

Siriapisith, T., Kusakunniran, W., and Haddawy, P. (2021). A 3D deep learning approach to epicardial fat segmentation in non-contrast and post-contrast cardiac CT images. PeerJ. Computer Science *7*, e806. https://doi.org/10.7717/peerj-cs.806.

Stringer, C., Wang, T., Michaelos, M., and Pachitariu, M. (2020). Cellpose: a generalist algorithm for cellular segmentation.

Sugawara, K., Çevrim, Ç., and Averof, M. (2022). Tracking cell lineages in 3D by incremental deep learning. Elife *11*. https://doi.org/10.7554/eLife.69380.

Wang, Z., and Wang, Z. (2020). A generic approach for cell segmentation based on Gabor filtering and area-constrained ultimate erosion. Artif. Intell. Med. *107*, 101929. https://doi.org/10.1016/j.artmed.2020.101929.

Wang, W., Taft, D.A., Chen, Y.-J., Zhang, J., Wallace, C.T., Xu, M., Watkins, S.C., and Xing, J. (2019). Learn to segment single cells with deep distance estimator and deep cell detector. Comput. Biol. Med. *108*, 133–141. https://doi.org/10.1016/j.compbiomed.2019.04.006.

Wang, Z., Bovik, A.C., Sheikh, H.R., and Simoncelli, E.P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. *13*, 600–612. https://doi.org/10.1109/tip.2003.819861.

Weigert, M., Schmidt, U., Boothe, T., Müller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S., et al. (2018). Content-aware image restoration: pushing the limits of fluorescence microscopy. Nat. Methods *15*, 1090–1097. https://doi.org/10.1038/s41592-018-0216-7.

Wen, C., Miura, T., Voleti, V., Yamaguchi, K., Tsutsumi, M., Yamamoto, K., Otomo, K., Fujie, Y., Teramoto, T., Ishihara, T., et al. (2021). 3DeeCellTracker, a deep learning-based pipeline for segmenting and tracking cells in 3D time lapse images. Elife *10*. https://doi.org/10.7554/eLife.59187.

Yang, L., Ghosh, R.P., Matthew Franklin, J., Chen, S., You, C., Narayan, R.R., Melcher, M.L., and Liphardt, J.T. (2020). NuSeT: A deep learning tool for reliably separating and analyzing crowded cells. PLoS Comput. Biol. *16*, 1–20. https://doi.org/10.1371/journal.pcbi.1008193.

Yang, Z., Wang, W., Zhao, L., Wang, X., Gimple, R.C., Xu, L., Wang, Y., Rich, J.N., and Zhou, S. (2021). Plasma cells shape the mesenchymal identity of ovarian cancers through transfer of exosome-derived microRNAs. Sci Adv *7*. https://doi.org/10.1126/sciadv.abb0737.

Zhang, A., Lipton, Z.C., Li, M., and Smola, A.J. (2021). Dive into Deep Learning.