Machine learning meets classical computer vision for accurate cell identification

Elham Karimi[*1], **Morteza Rezanejad**[*2,3], **Benoit Fiset**[1], **Lucas Perus**[4], **Sheri A. C. McDowell**[4], **Azadeh Arabzadeh**[1], **Gaspard Beugnot**[2,5] **Peter Siegel**[1,6,7], **Marie-Christine Guiot**[8], **Daniela F. Quail**[1,4,6], **Kaleem Siddiqi**[2], **Logan A. Walsh**[1,9]

[1]Rosalind and Morris Goodman Cancer Research Centre, McGill University, Montreal, QC, Canada

[2]School of Computer Science & Centre for Intelligent Machines, McGill University, Montreal, QC, Canada

[3]Departments of Psychology and Computer Science, University of Toronto, Toronto, ON, Canada

[4]Department of Physiology, Faculty of Medicine, McGill University, Montreal, QC, Canada

[5]Inria, École Normale Supérieure, CNRS, PSL Research University, Paris, France

[6]Department of Medicine, Division of Experimental Medicine, McGill University, Montreal, QC, Canada

[7]Department of Biochemistry, McGill University, Montreal, QC, Canada

[8]Department of Pathology, McGill University, Montreal, QC, Canada

[9]Department of Human Genetics, McGill University, Montreal, QC, Canada

[*]**Equal contribution/Co-first authors**

Correspondence:

Logan A. Walsh, Ph.D.
Assistant Professor, Department of Human Genetics, Rosalind and Morris Goodman Cancer Research Centre
McGill University
1160 Pine Ave
Montreal, QC, Canada H3A 1A3
p: 514-398-3052
e: logan.walsh@mcgill.ca

Kaleem Siddiqi, Ph.D.
Professor, School of Computer Science, Centre for Intelligent Machines
McGill University
3480 Rue University
Montreal, QC, Canada H3A 0E9
p: 514-398-3371
e. siddiqi@cim.mcgill.ca

Abstract

High-parameter multiplex immunostaining techniques have revolutionized our ability to image healthy and diseased tissues with unprecedented depth; however, accurate cell identification and segmentation remain significant downstream challenges. Identifying individual cells with high precision is a requisite to reliably and reproducibly interpret acquired data. Here we introduce CIRCLE, a cell identification pipeline that combines classical and modern machine learning-based computer vision algorithms to address the shortcomings of current cell segmentation tools for 2D images. CIRCLE is a fully automated hybrid cell detection model, eliminating subjective investigator bias and enabling high-throughput image analysis. CIRCLE accurately distinguishes cells across diverse tissues microenvironments, resolves low-resolution structures, and can be applied to any 2D image that contains nuclei. Importantly, we quantitatively demonstrate that CIRCLE outperforms current state-of-the-art image segmentation tools using multiple accuracy measures. As high-throughput multiplex imaging grows closer toward standard practice for histology, integration of CIRCLE into analysis protocols will deliver unparalleled segmentation quality.

Recent advances in multiplex imaging have revolutionized our ability to identify functionally meaningful cell populations among spatially and phenotypically heterogeneous tissues. Single-cell profiling using imaging mass cytometry (IMC), co-detection by indexing (CODEX), DNA-barcoding, serial immunostaining, or similar high-parameter imaging approaches have revealed that clonal evolution and spatially distinct tissue microenvironments drive cellular heterogeneity and control pathophysiology [1, 2, 3, 4, 5, 6, 7]. For example, the recent systematic, multi-dimensional interrogation of breast cancer using IMC revealed a detailed spatial map of single-cell phenotypes and cellular communities and resolved novel subtypes of breast cancer with distinct clinical outcomes [2]. Such multicellular spatial information provides a basis to study how geographical and phenotypic tissue features influence disease outcomes.

The most important requirement of image-based cellular profiling is the ability to identify individual cells through segmentation with extremely high precision, as this directly dictates the correctness and clinical relevance of all downstream discovery-based analyses. To do this, nuclei must be accurately distinguished, therefore the value of the technology hinges on accurate nuclear segmentation methodologies. This is a fundamental challenge in biomedical imaging primarily due to: (i) the inherently low contrast of edges and geometric boundaries, (ii) the lack of color in the image (most imaging technologies output in grayscale and use pseudo color), (iii) the lack of a unique defining texture, (iv) the varying shapes and sizes of cells, (v) the presence of overlapping cells in the image due to partial volume effects, (vi) the lack of ground truth examples, which limits the performance of machine learning methods (due to the inherent lack of training data), and (vii) the high density of cells, particularly in diseased tissue such as tumors. Each of these properties affects the design of computer vision algorithms to segment cell instances. It is also largely impossible to manually verify image quality in high-throughput experiments and there is a lack of automated methods to objectively flag or remove images and cells that are affected by artifacts. As a result, the existing tools for image-based cellular profiling regrettably require significant subjective manual intervention, which limits data reproducibility and reliability.

Over the years several methods have been developed for image-based cell identification, but each approach has limitations that are often overlooked. Image thresholding while adapting to local contrast [8] is very common. Here regions are distinguished by intensity averages or other measures of separation that are chosen either heuristically or intelligently. While most cell profiling techniques can benefit from this as an initial step, there is no guarantee that a distinct separation of cells can be achieved. Another classical method for medical image segmentation is the watershed algorithm, which equates pixel values to local topography [9, 10, 11]. Watershed based algorithms are widely used and are very effective when the regions to be segmented have homogeneous intensities. However, they often fail to differentiate between texture-based edges and true organelle boundaries. Active contour models [12, 13] allow geometric curves or surfaces to evolve and capture the boundaries of specific visual elements. They do so by minimizing a suitable energy functional, which incorporates both local (boundary) and global (region) image information so that the curves or surfaces converge to object boundaries [14, 15]. However, the outputs obtained by such models can be sensitive to the choice of initial seed contours, and separating overlapping cells can be a challenge. A further obstacle in using active contour models is the challenge of finding initial seed mask contours.

With advances in machine learning-based models, several promising approaches to image-based cell identification have been developed in both semi-supervised and supervised settings [16]. Many of these models use existing popular deep learning frameworks such as UNet [17] and MaskRCNN [18]. A general requirement of deep neural network (DNN)-based models is the availability of large annotated training datasets. Further, DNN models represent somewhat of a 'black box' for the user since they presently offer little insight for troubleshooting when a desired cell population is not accurately identified. Despite these limitations, DNN models can benefit from a concept referred to as "transfer learning" [19, 20, 21, 22]. When deep models are initially trained on huge datasets of generic objects they learn to extract certain high-level visual features that aid a recognition or categorization task. Their pre-trained weights can then be later fine-tuned on specific tasks, with smaller amounts of task-specific training data.

In recent years, additional supervised learning-based cell identification methods from data points that represent cell locations have been established. There are advantages to approaching cell identification by combining both region and boundary information. Advances in UNet-based models enable advanced cell identification mechanisms in which all pixels that are surrounded by a cell instance are marked and labeled separately. Among these methods, StarDist[23] is an effective UNet-based model that learns a shape representation for cells when it detects them in images. The UNet backbone contributes to excellent foreground detection[24] and as a result, StarDist has a high recall, successfully finding locations where candidate cells exist. On the other hand, the cell boundaries provided by StarDist are not always accurately placed. In particular, its precision suffers when the underlying images have a low signal-to-noise ratio. Another recent method which also uses deep neural networks, Cellpose [25], promotes the benefits of crowdsourcing to acquire labeled training datasets to gradually improve the performance of such methods.

Motivated by the strengths and weaknesses of the aforementioned strategies for image-based cell-profiling, we developed CIRCLE (Cell segmentatIon acRoss sCaLEs), a state-of-the-art framework that is capable of automated

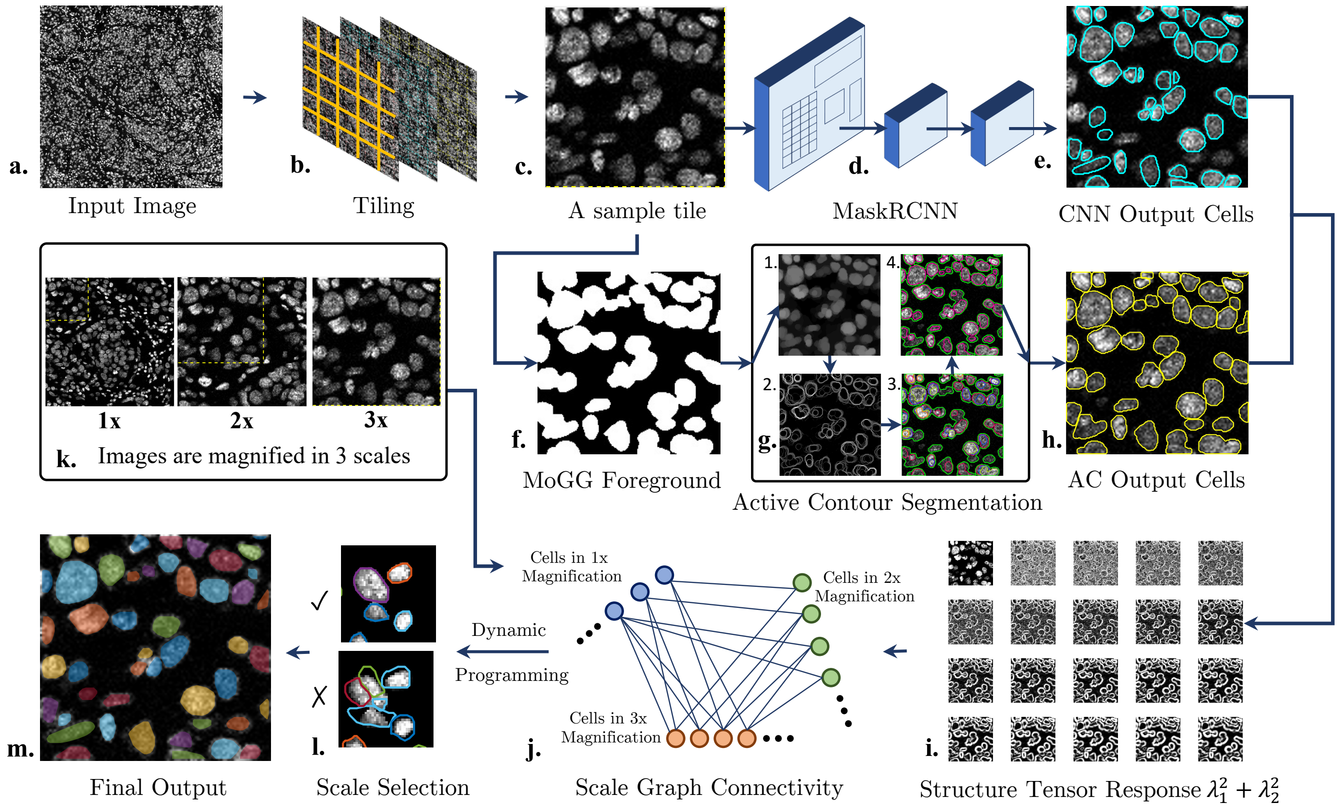


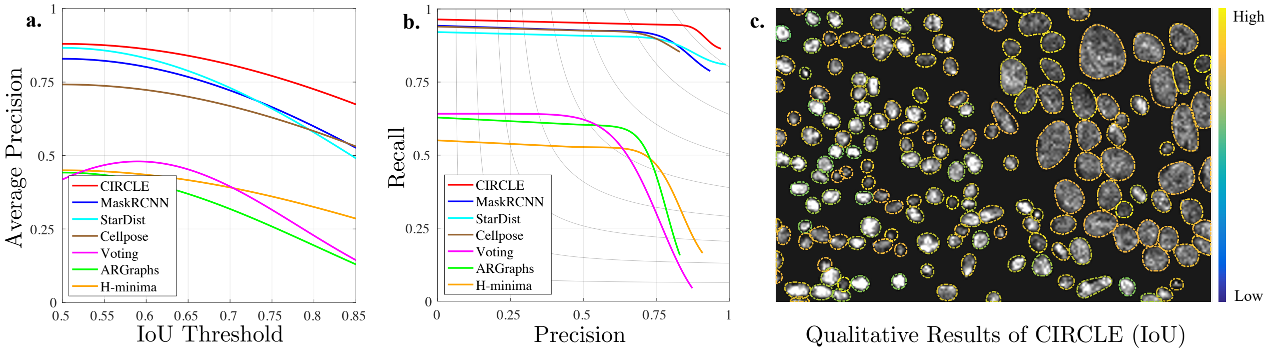
Figure 1: Schematic overview of the CIRCLE pipeline a. Representative input image (stained nuclei). **b.** Image is tiled, and shifted copies of the tiles are created so that cells lying at the borders are not missed. **c.** Each tile is processed at several magnification scales (represented in **k.**) allowing cells of varying sizes to be detected. **d-e.** For the deep model, MaskRCNN is used for instance cell segmentation. **f.** Concomitantly, tiles also serve as inputs for Mixture of Generalized Gaussian (MoGG) foreground mask generation. **g.** Active contour based refinement is applied to the MoGG foreground mask to yield a cell segmentation result for each tile that is independent of MaskRCNN **h.** The MaskRCNN and active contour output cells are pooled together. **i.** For each cell a structure tensor based feature is computed within a narrow band of each cell boundary. **j.** A scale graph is built to consider the structure tensor based feature across different magnification scales. **l.** Using dynamic programming, we select cells amongst the different magnifications. For overlapping detections, the best magnification is chosen and the other detections in the pooled results are removed. **m.** Representative image of the final output.

processing of high dimensional images with exceptional accuracy (Fig. 1). CIRCLE is a 2D cell identification pipeline that combines classical and modern machine learning-based computer vision algorithms to address shortcomings of commonly used image segmentation tools ([25, 18, 23, 10, 26, 9]). CIRCLE integrates a CNN for nuclear segmentation, with a novel combined active contour model and scale selection process. The CIRCLE pipeline can accommodate low-resolution images and is implemented efficiently.

We first divide the input image into smaller tiles that can each be analyzed separately. We detect cells in each of these smaller tiles at multiple magnifications, using both a deep neural network model and an active contour segmentation method. For the former process, we use the state-of-the-art Mask Region Convolutional Neural Network (MaskRCNN) architecture [18] to generate segmentation masks for each cell.

MaskRCNN provides high precision and requires minimal complex post-processing. The same image patch is also provided as input to a novel active contour segmentation method (ACSM). In the ACSM, the foreground regions are first found using a Mixture of Generalised Gaussians algorithm to include nuclei that could be potentially missed by the DNN model. We then apply a series of image processing operations to find all possible cells within these foreground regions. At the end of this process, we create a set of structure tensor (ST) maps to describe the intensity distribution in the vicinity of the estimated cell boundaries at each magnification scale. We then construct a connectivity graph across the detected cells at each scale. Using features derived from the ST, we apply a dynamic programming method to select the best scale for each cell, like the one that maximizes the ST feature response at the boundary. Scale selection enables our algorithm to avoid under-segmentation and over-segmentation of cells, even in very noisy conditions, and to handle cells of a variety of sizes.

To quantitatively compare the accuracy of CIRCLE against current state-of-the-art segmentation methods, first, we trained and fine-tuned the deep models CIRCLE, MaskRCNN, and StarDist on the 2018 Data Science Bowl (DSB 2018) competition dataset (https://www.kaggle.com/c/data-science-bowl-2018). Cellpose was pre-trained and fine tuned on multiple large datasets including the DSB 2018. Next, we created an entirely new test dataset, GCIMC, by meticulously manually segmenting four 1000 × 1000 pixel images generated through IMC of brain metastases from 4 patients. GCIMC contains **12692** segmented nuclei. When applied to the GCIMC dataset, CIRCLE was the most accurate segmentation method as determined by several measures, including 1) average precision across a range of intersection-over-union (IoU) thresholds (Fig. 2, a.), 2) precision versus recall (Fig. 2, b.) and 3) the Dice Coefficient (Fig. 2, d.). To qualitatively visualize the accuracy, dynamic range, and broad applicability of CIRCLE, we have included representative segmented IMC images from normal and cancerous tissues that exhibit various challenging features including poor contrast, varying cell size, and overlapping cells. (Fig. 3). These data highlight that although current methods may perform reasonably well on select tissues, CIRCLE uniformly is the most accurate, especially when confronted with heterogeneous and complex tissue microenvironments.



d.

Threshold τ	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
H-minima	58.63	58.28	57.46	56.64	55.35	54.08	52.83	51.59	50.37	35.37	$\boxed{18.82}$
ARGraphs	53.80	51.73	49.73	47.74	45.82	44.24	41.16	38.09	33.51	15.24	11.75
Voting	62.87	60.44	58.74	57.03	56.05	54.83	51.33	47.83	42.05	13.86	11.47
Cellpose	85.21	84.04	80.59	75.03	67.65	58.79	48.89	38.42	27.85	17.66	08.25
StarDist	$\boxed{90.20}$	$\boxed{88.78}$	$\boxed{86.81}$	$\boxed{84.44}$	81.07	76.44	69.66	59.02	43.64	32.98	15.37
MaskRCNN	87.87	87.07	86.06	84.67	$\boxed{82.86}$	$\boxed{80.67}$	$\boxed{76.97}$	$\boxed{71.01}$	$\boxed{60.51}$	$\boxed{42.38}$	18.00
CIRCLE	**93.72**	**92.72**	**92.14**	**91.04**	**90.56**	**88.87**	**87.64**	**85.25**	**83.24**	**79.22**	**69.15**

Figure 2: Quantitative comparison of CIRCLE with current state-of-the-art methods. We have applied CIRCLE to low-resolution multiplexed IMC data (GCIMC dataset) and have compared its accuracy to common segmentation algorithms. **a.** Average precision ($AP = \frac{TP}{TP+FP+FN}$). **b.** Precision versus Recall. **c.** Representative output of CIRCLE segmentation (boundaries are colored from blue to yellow according to their *several intersection over union* (IoU) values ranging from 0.5 to 1.0). **d.** Quantitative assessment of segmentation accuracy (Dice Coefficient, defined using precision (P) and recall (R) as $F_{\text{measure}} = \frac{2\times P\times R}{P+R}$ for IoU thresholds τ).

We are currently in an imaging revolution where images have become an unbiased source of quantitative information for biological phenotypes. As more studies exploit new technologies capable of generating highly multiplexed landscapes of tissue with spatial resolution, the need for accurate segmentation will only grow. In this setting, CIRCLE provides precise and automated cell identification while overcoming several limitations of other tools, offering a fully automated hybrid cell detection model to segment nuclei in 2D images.

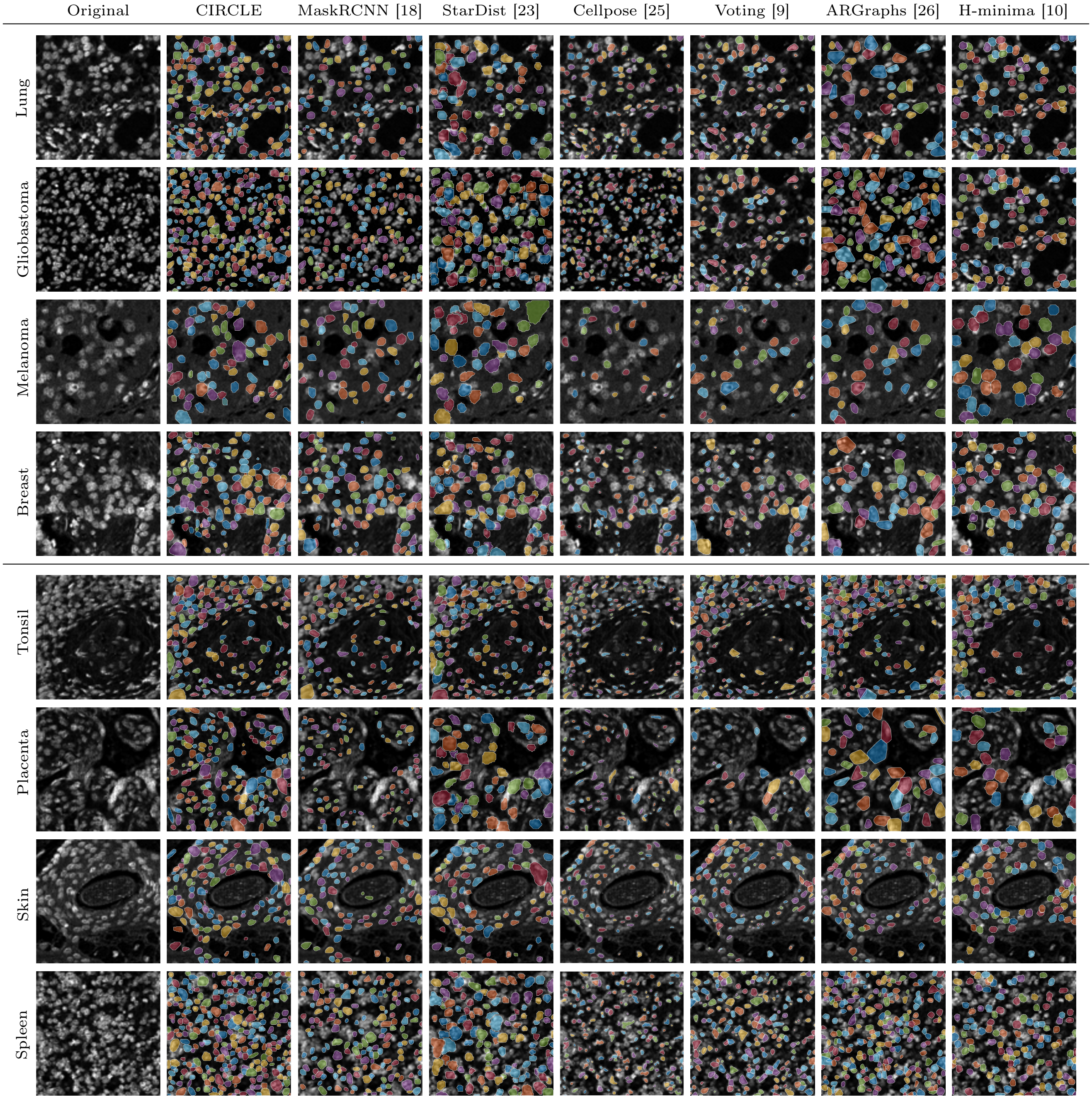


Figure 3: Qualitative comparison of CIRCLE with current state-of-the-art methods. Representative IMC images from normal and cancer tissues, reflecting a range of challenging cases including poor contrast, varying cell size, and overlapping cells, segmented with the indicated methods.

Methods

We describe a number of components of our workflow in the subsections below. These methods are combined in the manner illustrated in Figure 1. The general idea is to consider both the output of a powerful CNN-based segmentation method (Section 2) and one that uses a foreground detection approach first (Section 3), followed by an active contour refinement stage to separate potentially overlapping cells (Section 4). The two independent sets of candidate cell segmentations are then pooled and among these a selection criterion is applied across different magnification scales to select the best scale for those detections that have a spatial overlap (Section 5). This scale selection process allows for variation in cell size.

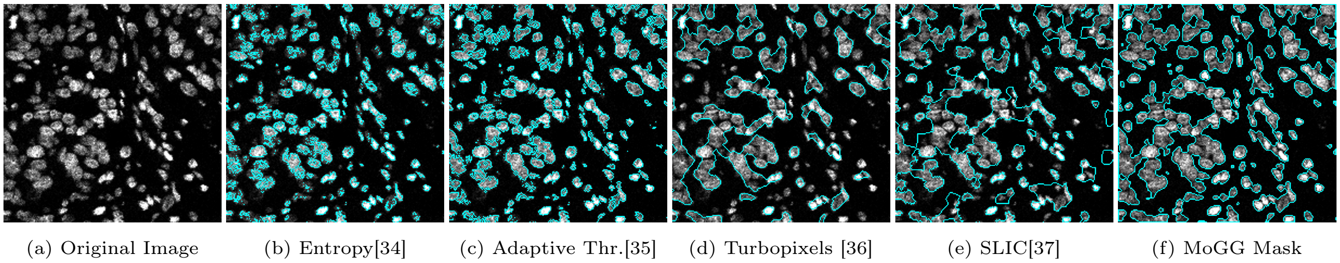
1. Tiling

To make our approach memory efficient, we divide each input image into a smaller set of tiles. The tiles are chosen to be larger than the maximum possible size of a cell to be detected. Tiling facilitates the use of standard CNN implementations, which typically assume that the input images are small, and mitigates the effect of non-uniform luminance across larger windows. Assuming that the source image is of size $W \times H$ and that the tile width is w_s, we partition the image into a grid with $m \times n$ tiles, where: $m = \left\lceil \frac{W}{w_s} \right\rceil$ and $n = \left\lceil \frac{H}{w_s} \right\rceil$. The tile width parameter is chosen empirically, e.g., for the images that are 1000×1000 in size, we chose w_s to be 200 to yield 200 tiles. Each tile is provided as a separate input to our method. All the detected cells that intersect a tile boundary are discarded. To ensure that no cells are missed we create two copies of the original tiling, shifting it by $\leq \frac{w_s}{2}$ to the South and to the East, and use these shifted tiles as inputs as well. The choice of w_s should be no smaller than the maximum possible size of a cell. To avoid duplicate detections of the same cell due to the additional shifted tiles, we maintain an occupancy image, which is the size of the original image, and fill it with the area that each detected cell covers.

2. MaskRCNN Cell Detection

To obtain one set of candidate cells, we apply the Mask Region Convolutional Neural Network (MaskRCNN) architecture to segment cell nuclei [18] (Fig. 1d-e). This network provides advantages over the popular UNet approach [17], having evolved through four different earlier versions. The latest version uses an RoIAlign layer to correct for misalignments and generates both the bounding boxes and segmentation masks for each cell in the image. MaskRCNN is already used [27, 28, 29, 30, 31, 32] to segment nuclei/cells or lesions in medical imaging. We used the existing model, pre-trained on the COCO dataset [33], as a starting point and then fine-tuned it on the data science bowl public dataset available at `https://www.kaggle.com/c/data-science-bowl-2018`. We applied the various standard data augmentation techniques including both proportional and non-proportional image resizing, random rotations in increments of 10 degrees (36 rotations in all), and vertical and horizontal flips. The fine-tuning consists of training the network in 3 batches of 25 epochs each, where each batch's trained weights were used as initial weights for the next batch, and the learning rates were reduced for the subsequent batch.

3. Foreground Detection


(a) Original Image (b) Entropy[34] (c) Adaptive Thr.[35] (d) Turbopixels [36] (e) SLIC[37] (f) MoGG Mask

Extended data Figure 1. A comparison of foreground detection approaches on a typical tissue sample. In general we found the MoGG method to be robust to non-uniformities in luminance and degradation in image quality, and to have a high ratio of true positives to false negatives (missed foreground detections).

Foreground detection refers to the marking of any region in the image that could potentially contain a cell. Whereas a naive approach would involve intensity-based thresholding, more sophisticated methods use histograms [38, 39, 40, 41, 42], clustering [43, 44, 39, 45], or the entropy of foreground and background regions [46, 47, 48, 49, 34]. Since our main purpose is to mark regions whose boundaries will be later refined, a multiclass histogram thresholding approach to efficiently model multi-modal class-conditional distributions using mixtures of generalized Gaussian distributions (MoGG) [50] works well, in comparison to alternative approaches including an entropy-based method [34], the use of an adaptive threshold [35], and the use of superpixels for oversegmentation [36, 37] (Extended data Figure 1). MoGG provides a flexible and suitable tool for various problems in image segmentation including ones that arise in medical image analysis [51, 52, 53, 54, 55]. The foreground detection is an essential step since it improves recall by adding additional regions that may have been missed by the neural network. These foreground regions are refined using an active contour model to provide candidate cell segmentations, as described next.

4. Active Contour Based Cell Segmentation

The output of the MoGG foreground detection method is now refined to better separate potentially overlapping cells. We exploit the property that when two or more convex objects overlap in the 2D plane, there are two local peaks in the distance function from their combined boundary.

4.1. Morphological Opening

Morphological opening consists of an erosion step followed by a dilation step [56] to remove structures smaller than the diameter of the structuring element: $I_o = I \circ s = (I \ominus s) \oplus s$. Here I is the input image patch, and s is the structuring element. We use a circular structuring element with a diameter d chosen to be linearly proportional to the considered scale. The result of this opening operation on the sample tile is shown in Figure 1g, panel 1. The opened image results in interior cell regions with a more homogeneous appearance.

4.2. Gradient Rings

We now devise a way to separate overlapping cells in the opened image. We use the magnitude of the gradient within the opened image as a measure of cell boundary strength (Fig. 1g, panel 2): $\|\nabla I_o\| = \sqrt{(\frac{\partial I_o}{\partial x})^2 + (\frac{\partial I_o}{\partial y})^2}$ where $\frac{\partial I_o}{\partial x}$ and $\frac{\partial I_o}{\partial y}$ are the partial derivatives of the image in the x and y directions, respectively. We then consider level sets of the gradient magnitude, normalized to lie between 0 and 1, to provide a series of nested gradient rings, as illustrated in Figure 1g, panel 3.

4.3. Generating Seed Contours

We now process the nested gradient rings in order to provide seed contours to segment individual cells. Each gradient ring, corresponding to a simple closed contour, is treated separately (as illustrated using different colors in Fig. 1g, panel 3). We denote the i^{th} ring by r_i and prune rings in a recursive manner. Here, ring r_i is removed if there is another ring r_j such that $r_i \subseteq r_j$ and no other ring r_k exists such that $r_k \subseteq r_j$. The pruning stops when no further rings can be removed. We verify the $r_i \subseteq r_j$ condition by checking whether the area bounded by r_i is contained within the area bounded ring r_j. The surviving rings are shown in red in Figure 1g, panel 4.

4.4. Active Contour Cell Segmentation

In a final step, we refine the surviving rings to segment likely cell boundaries within an image tile, using the Chan-Vese region-based active contour [57]. This formulation best separates each MoGG foreground region into two classes, under the assumption that each class is homogeneous in intensity and thus well represented by its mean. Here, we assume a system of multiple active contours that evolve simultaneously at each time step. Let r represent a ring, and u_0 represent the set of pixels within the same MoGG foreground region, excluding r. The algorithm evolves the contour to minimize the following energy function, ultimately placing the evolved contour at the boundary of two classes:

$$\begin{aligned}F(c_1, c_2, r_c) = &\mu \times \text{Length}(r) + \nu \times \text{Area}(\mathit{interior}(r)) \\ &+ \lambda_1 \int_{\mathit{inside}(r_c)} |u_0(x,y) - c_1|^2 \partial x \partial y \\ &+ \lambda_2 \int_{\mathit{exterior}(\bigcup r_i)} |u_0(x,y) - c_2|^2 \partial x \partial y.\end{aligned} \tag{1}$$

Here $\mathit{interior}(r)$ refers to the pixels contained within r, and $\mathit{exterior}(\bigcup r_i)$ is the set of all pixels in the foreground region excluding those associated with all present rings. c_1 is the mean intensity of u_0 inside r and c_2 is the mean intensity of u_0 in the region outside r, within the foreground region. $\mu \geq 0$, $\nu \geq 0$, and $\lambda_1, \lambda_2 > 0$ are empirically chosen fixed weight parameters. Using a level set formulation [58, 59, 60], we evolve the ring r so as to minimize the above energy. Each ring is updated using the following level set equation at each time step:

$$\frac{\partial \Phi}{\partial t} = \delta(\Phi)\Big[\mu \nabla . \frac{\nabla \Phi}{|\nabla \Phi|} - \nu - \lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2\Big], \tag{2}$$

where δ is the Dirac-delta function. Here, $\Phi(x, y, r)$ is the embedding function whose level sets represent the evolving rings [58, 60, 59]. With multiple level sets evolving simultaneously, the pixels that are inside and outside of rings within u_0 are updated in time. For the level set update we use the numerical method in [61]. The resulting detected cells are shown in Figure 1h.

5. Scale Selection

We now have two independent sets of candidate cells at each magnification scale: those obtained by the CNN (Fig. 1e) and those obtained by the active contour detection process (Fig. 1h). We have devised a method to select between these candidate detections over the considered scales. We first compute a salience score for each cell at each magnification scale. To do so, for each magnification, we blur the image patch with Gaussians over a range of blurring scales. We then compute the Structure Tensor (ST) for all standard deviations at each pixel. The ST is computed at each pixel $\mathbf{p}$ as:

$$ST_s(\mathbf{p}, \sigma) = \begin{bmatrix} G_\sigma * I_x(\mathbf{p})^2 & G_\sigma * I_x(\mathbf{p}) I_y(\mathbf{p}) \\ G_\sigma * I_x(\mathbf{p}) I_y(\mathbf{p}) & G_\sigma * I_y(\mathbf{p})^2 \end{bmatrix}, \tag{3}$$

where $I_x(\mathbf{p})$ and $I_y(\mathbf{p})$ are partial derivatives in space in the directions x and y of the image I, computed using central differences. G_σ is a Gaussian filter of length $2\sigma + 1$ with a standard deviation of $\frac{\sigma}{k}$, where k is an arbitrary positive constant.

We then compute a local feature descriptor measure using the sum of the squared eigenvalues of the ST matrix:

$\Lambda(\mathbf{p}, \sigma) = \lambda_1^2 + \lambda_2^2$. In principle, we expect $\lambda_1^2 + \lambda_2^2$ to be relatively large at a location near the cell boundary or when when two cell boundaries are touching each other (Fig. 1 (i)). For each scale s let n_s be the number of pooled CNN and active contour cell detections, stored in a list $\mathcal{C}_s = \{c_{1,s}, \ldots, c_{n_s,s}\}$. We define a salience score for each cell as the highest average value of $\Lambda(\mathbf{p})$ within the vicinity of a boundary of a cell, amongst all possible standard deviations as

$$\zeta(c_{i,s}) = \underset{\sigma}{\operatorname{argmax}} \frac{\sum_{\mathbf{p} \in B(c_{i,s})} \Lambda(\mathbf{p})}{\#\{B(c_{i,s})\}}.$$

Here $B(c_{i,s})$ is a one-pixel expansion of the cell boundary $c_{s,i}$ towards the inside and outside. We accommodate for size variation in cell nuclei by considering 3 different magnification scales (Fig. 1k). To select between these magnification scales and also the different cell generation methods, we construct a connectivity graph across cells and magnification scales. Each cell at each magnification scale is a node of the connectivity graph, with a weight of $c_{i,s}$. We connect two cells c_{i,s_1} and c_{j,s_2} if the ratio of the common area between those two cells over the union of the areas of these cells is at above a certain threshold:

$$\tau \leq \frac{A(c_{i,s_1} \cap c_{j,s_2})}{A(c_{i,s_1})} \text{ or } \tau \leq \frac{A(c_{i,s_1} \cap c_{j,s_2})}{A(c_{i,s_2})}.$$

Here $A(c_{k,m})$ represents the area of the k^{th} cell at magnification m and τ is a particular threshold chosen between 0 to 1. In our experiments, we chose τ to be 0.85. After the entire connectivity graph is constructed (Fig. 1j), we update each node's $\zeta(c_{s,i})$ score by dividing it by the degree of that node: $\zeta(c_{s,i}) \leftarrow \frac{\zeta(c_{s,i})}{\deg(c_{s,i})}$.

Following the construction of the connectivity graph, we select cells using a dynamic programming algorithm that maximizes the sum of $\zeta(c_{s,i})$ scores over all identified cells. Let $\mathcal{C}_i$ represent the cells at i^{th} magnification level. Our goal is to optimize an energy function $\mathcal{U}(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3)$ that selects cells between these magnification scales and the two detection methods, while avoiding overlapping detections. For this we construct the following energy

$$\mathcal{U}(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3) = \max_{i=1,2,3} \begin{cases} \mathcal{U}(\mathcal{C}_1 - \{c_{i,1}\}) + \zeta(c_{i,1}) \\ \mathcal{U}(\mathcal{C}_2 - \{c_{i,2}\}) + \zeta(c_{i,2}) \\ \mathcal{U}(\mathcal{C}_3 - \{c_{i,3}\}) + \zeta(c_{i,3}) \end{cases}, \tag{4}$$

where $\mathcal{C}_j - \{c_{i,j}\}$ refers to the removal of cell $c_{i,j}$ and its connections from all cell sets $\mathcal{C}_i$. As the connections between overlapping cells from the two different methods and the different magnifications scales are sparse, we used the Cuthill-McKee algorithm [62] to permute the sparse matrix of connections, for increased computational efficiency.

Data and Code Availability

All data and source code will be made available without restrictions. Ethics approval IRB00010120. Normal tissues were de-identified excess diagnostic tissue controls that were slated for incineration.

References

[1] Liudahl, S. M., Betts, C. B., Sivagnanam, S., Morales-Oyarvide, V., da Silva, A., Yuan, C., Hwang, S., Grossblatt-Wait, A., Leis, K. R., Larson, W., et al. *Cancer Discovery* (2021).

[2] Jackson, H. W., Fischer, J. R., Zanotelli, V. R., Ali, H. R., Mechera, R., Soysal, S. D., Moch, H., Muenst, S., Varga, Z., Weber, W. P., et al. *Nature* **578**(7796), 615–620 (2020).

[3] Schürch, C. M., Bhate, S. S., Barlow, G. L., Phillips, D. J., Noti, L., Zlobec, I., Chu, P., Black, S., Demeter, J., McIlwain, D. R., et al. *Cell* **182**(5), 1341–1359 (2020).

[4] Ji, A. L., Rubin, A. J., Thrane, K., Jiang, S., Reynolds, D. L., Meyers, R. M., Guo, M. G., George, B. M., Mollbrink, A., Bergenstråhle, J., et al. *Cell* **182**(2), 497–514 (2020).

[5] Damond, N., Engler, S., Zanotelli, V. R., Schapiro, D., Wasserfall, C. H., Kusmartseva, I., Nick, H. S., Thorel, F., Herrera, P. L., Atkinson, M. A., et al. *Cell metabolism* **29**(3), 755–768 (2019).

[6] Porpiglia, E., Samusik, N., Van Ho, A. T., Cosgrove, B. D., Mai, T., Davis, K. L., Jager, A., Nolan, G. P., Bendall, S. C., Fantl, W. J., et al. *Nature cell biology* **19**(5), 558–567 (2017).

[7] Ali, H. R., Jackson, H. W., Zanotelli, V. R., Danenberg, E., Fischer, J. R., Bardwell, H., Provenzano, E., Rueda, O. M., Chin, S.-F., Aparicio, S., et al. *Nature Cancer* **1**(2), 163–175 (2020).

[8] Nawandhar, A. A., Yamujala, L., and Kumar, N. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1199–1203, (2015).

[9] Xu, H., Lu, C., and Mandal, M. *IEEE journal of biomedical and health informatics* **18**(5), 1729–1741 (2014).

[10] Koyuncu, C. F., Akhan, E., Ersahin, T., Cetin-Atalay, R., and Gunduz-Demir, C. *Cytometry Part A* **89**(4), 338–349 (2016).

[11] Meyer, F. *Signal processing* **38**(1), 113–125 (1994).

[12] Malladi, R., Sethian, J. A., and Vemuri, B. C. *IEEE transactions on pattern analysis and machine intelligence* **17**(2), 158–175 (1995).

[13] Kass, M., Witkin, A., and Terzopoulos, D. *International journal of computer vision* **1**(4), 321–331 (1988).

[14] Caselles, V., Kimmel, R., and Sapiro, G. *International journal of computer vision* **22**(1), 61–79 (1997).

[15] Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., and Yezzi, A. In *Proceedings of IEEE International Conference on Computer Vision*, 810–815. IEEE, (1995).

[16] Moen, E., Bannon, D., Kudo, T., Graf, W., Covert, M., and Van Valen, D. *Nature methods* **16**(12), 1233–1246 (2019).

[17] Ronneberger, O., P.Fischer, and Brox, T. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, 234–241. Springer, (2015). (available on arXiv:1505.04597 [cs.CV]).

[18] He, K., Gkioxari, G., Dollár, P., and Girshick, R. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969, (2017).

[19] George Karimpanal, T. and Bouffanais, R. *Adaptive Behavior* **27**(2), 111–126 (2019).

[20] Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., and Zhang, G. *Knowledge-Based Systems* **80**, 14–23 (2015).

[21] Bozinovski, S. *Informatica* **44**(3) (2020).

[22] Pratt, L. Y. *Advances in neural information processing systems* , 204–204 (1993).

[23] Schmidt, U., Weigert, M., Broaddus, C., and Myers, G. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 265–273. Springer, (2018).

[24] Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., and Maier-Hein, K. H. *Nature methods* **18**(2), 203–211 (2021).

[25] Stringer, C., Wang, T., Michaelos, M., and Pachitariu, M. *Nature Methods* **18**(1), 100–106 (2021).

[26] Arslan, S., Ersahin, T., Cetin-Atalay, R., and Gunduz-Demir, C. *IEEE transactions on medical imaging* **32**(6), 1121–1131 (2013).

[27] Johnson, J. W. *CoRR* **abs/1805.00500** (2018).

[28] Anantharaman, R., Velazquez, M., and Lee, Y. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2197–2204. IEEE, (2018).

[29] Vuola, A. O., Akram, S. U., and Kannala, J. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 208–212. IEEE, (2019).

[30] Fujita, S. and Han, X.-H. In *Proceedings of the Asian Conference on Computer Vision*, (2020).

[31] Durkee, M. S., Sibley, A., Ai, J., Abraham, R., Liarski, V. M., Clark, M. R., and Giger, M. L. In *Medical Imaging 2020: Digital Pathology*, volume 11320, 1132019. International Society for Optics and Photonics, (2020).

[32] Moshkov, N., Mathe, B., Kertesz-Farkas, A., Hollandi, R., and Horvath, P. *Scientific reports* **10**(1), 1–7 (2020).

[33] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. In *European conference on computer vision*, 740–755. Springer, (2014).

[34] Brink, A. D. and Pendock, N. E. *Pattern recognition* **29**(1), 179–188 (1996).

[35] Bradley, D. and Roth, G. *Journal of graphics tools* **12**(2), 13–21 (2007).

[36] Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., and Siddiqi, K. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(12), 2290–2297 (2009).

[37] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. *IEEE transactions on pattern analysis and machine intelligence* **34**(11), 2274–2282 (2012).

[38] Weszka, J. S. and Rosenfeld, A. *IEEE Transactions on Systems, Man, and Cybernetics* **8**(8), 622–629 (1978).

[39] Rosenfeld, A. and De La Torre, P. *IEEE Transactions on Systems, Man, and Cybernetics* (2), 231–235 (1983).

[40] Sezan, M. I. *Computer vision, graphics, and image processing* **49**(1), 36–51 (1990).

[41] Ramesh, N., Yoo, J.-H., and Sethi, I. *IEE Proceedings-Vision, Image and Signal Processing* **142**(5), 271–279 (1995).

[42] Guo, R. and Pandit, S. *Machine Vision and Applications* **10**(5), 331–338 (1998).

[43] Olivo, J.-C. *CVGIP: Graphical Models and Image Processing* **56**(3), 205–218 (1994).

[44] Otsu, N. *IEEE transactions on systems, man, and cybernetics* **9**(1), 62–66 (1979).

[45] Jawahar, C., Biswas, P. K., and Ray, A. *Pattern Recognition* **30**(10), 1605–1613 (1997).

[46] Kapur, J. N., Sahoo, P. K., and Wong, A. K. *Computer vision, graphics, and image processing* **29**(3), 273–285 (1985).

[47] Yen, J.-C., Chang, F.-J., and Chang, S. *IEEE Transactions on Image Processing* **4**(3), 370–378 (1995).

[48] Pun, T. *Signal processing* **2**(3), 223–237 (1980).

[49] Pun, T. *Computer graphics and image processing* **16**(3), 210–239 (1981).

[50] Boulmerka, A., Allili, M. S., and Ait-Aoudia, S. *Pattern Recognition* **47**(3), 1330 − 1348 (2014). Handwriting Recognition and other PR Applications.

[51] Nacereddine, N., Tabbone, S., Ziou, D., and Hamami, L. In *2010 20th International Conference on Pattern Recognition*, 4557–4560. IEEE, (2010).

[52] Elguebaly, T. and Bouguila, N. *Signal Processing* **91**(4), 801–820 (2011).

[53] Elguebaly, T. and Bouguila, N. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, 207–218. Springer, (2010).

[54] Bouman, C. and Sauer, K. *IEEE Transactions on image processing* **2**(3), 296–310 (1993).

[55] Bazi, Y., Bruzzone, L., and Melgani, F. *Pattern Recognition* **40**(2), 619–634 (2007).

[56] Serra, J. *Signal processing* **38**(1), 3–11 (1994).

[57] Chan, T. F. and Vese, L. A. *IEEE Transactions on image processing* **10**(2), 266–277 (2001).

[58] Osher, S. and Sethian, J. A. *Journal of computational physics* **79**(1), 12–49 (1988).

[59] Osher, S. and Fedkiw, R. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, (2006).

[60] Sethian, J. A. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge university press, (1999).

[61] Khandelwal, P. *Spine Segmentation in Computed Tomography Images using Geometric Flows and Shape Priors*. PhD thesis, McGill University Libraries, (2019).

[62] Cuthill, E. and McKee, J. In *Proceedings of the 1969 24th national conference*, 157–172, (1969).