Combined Method of Deep Learning, Watershed Algorithm and Thresholding Algorithm for Cell Segmentation and Tracking

Con Tieu-Vinh, Minrui Lu, Shuang Liang, Xiaocong Chen, Yunfan Wang

***Abstract*—Automatic tracking and analysis of biological cells across time could be a powerful method for biological researches. There are two main challenges in this task. The first one is cell segmentation since an accurate cell contour is the foundation for cell detection. The other one is cell tracking, which would affect the outcome of cell analysis directly. This paper presents an approach of combining deep learning techniques and traditional computer vision methods for cell segmentation and tracking cell behaviors based on cell topology. To improve the accuracy of cell segmentation, we propose to preprocess the datasets exhibiting unclear and irregular cell boundaries using the deep learning frameworks of J-net or DeepWater first, and then use morphological methods to obtain a better cell segmentation. We compare the results of these frameworks to our own thresholding-based algorithm. To track cells' trajectory, we choose the distance of cell centroid movement and the area of cells as features in sequential images to judge whether cell motion or mitosis occurs. Finally, we analyze the frames in a certain period of time in backward order to get a cell's trajectory. Our methods are tested by using three datasets with different features, which are published on the Cell Tracking Challenge (CTC). These datasets are evaluated to have proper performance in detecting cell motion and mitosis.**

The Github repository of this paper is given below: *https://github.com/raymondluk1995/COMP9517* (available from Aug 8^{th}, 2020)

***Index Terms*—cell segmentation, cell tracking, mitosis detection, cell motion analysis**

I. Introduction

A good understanding of living things needs various analyses of its anatomic and dynamic properties, which can be gained from live-cell imaging experiments [1]. The more data these experiments generate, the better the behaviour of cell populations can be represented. Therefore, for better results, these experiments often produce a great volume of time-lapse image data. High-throughput spatio-temporal measurements assist us to detect cell behaviors like migration, mitosis, apoptosis and the reconstruction of cell lineages [2, 3], but is computationally complex. Other challenges are related to the quality and content of the image data like poor contrast with high noise levels, irregular cell contours, entry and exit of the cells [3, 4]. To deal with large amounts of data differing in quality and content, people have developed many kinds of cell detecting and tracking techniques.

For detecting, the easiest way is to use thresholding if cells in images have different intensities from their neighbouring areas [1]. However, due to its simplicity, it is one of the most error-prone methods and often fails when noise levels are high, contrast is low or cells are overlapping. A complex and popular method for detecting and segmentation is template matching, but it only performs well when cell contours are simple and regular [5]. Another popular and more robust approach is watershed transformation, which can completely separates cells but is prone to over-segmentation. Model-based segmentation can produce more sensible results, but is prone to under-segmentation. Therefore, both approaches mentioned above benefit from the application of post-processing techniques [6].

As for tracking techniques, finding the nearest cell in the previous frame based on the target cell's centroid position is a straightforward approach. It works well if cells in the images move at a slow speed and are not close to other cells. In the cases that cells are intensely populated or move quickly, this method should be extended to include some other basic features such as areas, contours, overall mean distances, curvature and intensities [6]. But even so, it still can not deal with the movement and apoptosis of some cells. In 2013, the rank-based filtering mechanism are refined in maximum cardinality minimum weight bipartite matching [3]. The matched pairs are sorted based on their matching cost and only the top q percentage of the pairs are selected as correct matches, which can deal with the exit or apoptosis of the cells with a higher probability. Another trend intending to use stochastic knowledge is called the probabilistic method, like Bayesian inference. They are also powerful techniques for tracking cells.

Our task is to try as many computer vision methods as possible, to compare their differences and to select the most appropriate way to detect, track and analyse cells. We have three data sets with different qualities and face three kinds of challenges. Firstly, cells in DIC-C2DH-HeLa set are comparatively larger and closer to each other that it is difficult to detect them and make a segmentation. We use two different methods to generate masks for later segmentation. The first method is called J-net, a multi-resolution neural network for semantic segmentation. To improve segmentation of the masks generated by this method, we then use some basic image processing operations, like thresholding, Gaussian blur, erosion, dilation and watershed algorithm, to improve the segmentation quality. The second method we apply is the DeepWater model, which can deal with this data set in an ideal way. Cells in Fluo-N2DL-HeLa set have low contrast and need some basic image pre-processing strategies. We use Gaussian blur to remove noises in images and use thresholding

to increase the contrast for better visualization. After that, we use erosion and dilation to separate the cells. Cells in the PhC-C2DL-PSC set are intensively scattered and some of them move in a high speed. Initially, we use the erosion operator and Gaussian blur to remove noise. Then we use thresholding to increase the contrast. This data set can also be dealt with the DeepWater model mentioned above. As for the tracking part, we adopt a nearest-neighbor linking approach based on the distances and areas of computed cells, which we find effective in matching most cells correctly across time.

In summary, the contribution of this paper includes:

- We propose a model that can deal with all three different kinds of data sets.
- We detect all the cells and draw the trajectory of each of them, and count the number of the cells.
- We detect the dividing cells and change the color of their contours, and count their number.
- We compute some properties for a selected cell, including its speed, total distance as well as net distance travelled up to that time point, and confinement ratio of its motion.
- We allow a user to retrieve specific frames with annotations displaying the above information.

II. Literature Review

A. Cell segmentation in computer vision.

Cell segmentation distinguishes individual cells from images and could be regarded as the most significant and the most fundamental task in cell analysis. The reason is that subsequent cell analysis, such as mitosis and motion, often require cell boundaries as an essential feature to do further processes [7].

In recent years, various state-of-the-art approaches have been proposed to improve cell segmentation. Traditional methods such as thresholding, region growing, seeded watershed segmentation, and edge-base segmentation [14] mainly focus on features of images. For instance, Chen et al. proposed a method to segment cells by performing global thresholding based on Otsu's algorithm [12]. However, this method only has an excellent performance on images with separated cells, not for those in which cell contours touch or overlap with others. Wählby et al. proposed an edge detection method based on the region, in which they use the gradient magnitude of objects pixels and background pixels in the image to apply watershed segmentation [13]. The results of these methods are, relatively, prone to be influenced by noise and oversegmentation [6]. More sophisticated methods often apply deep learning technologies to improve the accuracy of cell segmentation [8]. For example, Ronneberger et al. proposed a U-Net architecture based on annotated datasets processed with data augmentation and used an autoencoder to reconstruct images to obtain neuronal structures segmentation [9]. Van Valen et al. proposed a framework named DeepCell, which uses optimized deep convolutional neural networks to segment biological images and could be performed on different kinds of biological cell types [10]. Nath et al. proposed a time-saving approach based on four or fewer level set algorithm to segment moving epithelial cells [11]. However, the performance of the approaches mentioned above precisely depends on the types and shapes of cell datasets we would process. Hence, it is unrealistic to achieve proper cell segmentation by only one method, and we propose to combine deep learning approaches and traditional computer vision methods to segment the CTC cell datasets.

B. Cell tracking in computer vision.

Researching cells' activities like moving, dividing, and their health conditions in a certain period manually is extremely time-consuming and inefficient. In recent years, experts in the computer vision area have proposed methods to automatically track and analyze cell activities. These approaches can be presented in three broad categories [15]. The first is establishing a tracking model to detect a cell's path. For example, based on the mean shift algorithm, Debeir et al. proposed to build a cell tracking path by Vitro phase-contrast in video [16]. Ray et al. proposed tracking cells automatically by combining active contour and Kalman filter [17]. One tricky thing is that the fundamental structure of those model evaluation methods above cannot be used in mitosis directly [15], which means that further image processing is needed to improve tracking results. The second category is based on the result of cell segmentation. For example, Yan et al. proposed to use a classic watershed transform algorithm for cell segmentation and then use cell distance and sizes to track their path [18]. The tracking accuracy of this method depends large upon correct segmentation. The last group is a framework based on Bayesian probability. Kachouie et al. proposed a probabilistic model-based cell tracking method to locate separated cells [5]. In this method, one problem is that it does not have enough assumptions to establish accurate models for different cells [15]. Inspired by the approaches above, we propose to track cells based on segmentation and use topological features such as the distance between cells and their areas to identify cell activities.

III. Methods

A. Cell Segmentation

1) Overview: The proposed network combines two existing network designs. The first is the DeepWater network [19] which is a combination of a convolutional neural network and a marker-controlled watershed segmentation. The other one is J-net [20], a multi-resolution neural network for semantic segmentation. We apply the DeepWater network to the DIC-C2DH-HeLa dataset and the PhC-C2DL-PSC dataset, and the J-net network to the DIC-C2DH-HeLa dataset. Moreover, we develop a thresholding-based algorithm which incorporates Gaussian blurring, histogram normalization and morphological operations for the segmentation of Fluo-N2DL-HeLa dataset and PhC-C2DL-PSC dataset. Comparison of segmentation performance between different methods will be discussed in the experiment section.

2) DeepWater Network: In DeepWater network, two convolutional neural networks are trained. One (CNN_m) is for cell marker prediction, and the other one (CNN_c) is for image foreground (cell regions) prediction. With the outputs of the two CNN, marker-controlled watershed transformation is applied to generate the final segmentation. Both CNNs have the same structure. Each network is made of 18 convolutional layers with kernel size 3×3. Hour-glass topology with skip connections is applied to the design. The last convolutional layer has a kernel of size 1×1 and a soft-max activation function. A weighted cross-entropy loss function

$$L(p, y) = -\frac{\sum_{q \in \Omega} w(q) \log \left(p_{y(q)}(q)\right)}{\sum_{q \in \Omega} w(q)} \tag{1}$$

where w is a pixel weight function that is unique for every training sample is used in the network design. Assume we have a cell mask ϕ and a set of masks of all cells Φ, to each pixel q we have in the image, we assign a weight $w(q) \in \mathbb{R}^+$ by the formula:

$$w(q) = \left[1 + a \sum_{\phi \in \Phi} \max(d - \|q, \phi\|, 0)\right] \cdot b \tag{2}$$

where $||q, \phi||$ is the Euclidean distance from q to the closest pixel ϕ.

With predictions from CNN_m, we treat them as markers and put them into the watershed transformation. Watershed only focuses on pixels within the cell regions. The segmentation function which distinguishes the predictions from CNN_m and CNN_c controls the segmentation process. In the final segmentation, only one segment is associated with one marker.

3) J-net: J-net is a simplified version of U-net [21], which merely includes the expansive path. This network assembles the second half of U-net, therefore, it is named after its shape. In the design of J-net for the DIC-C2DH-HeLa dataset, three segments are included. Each segment is a combination of a CNN whose convolutional layers have 3×3 filters, including a deconvolution layer [22],which upsamples the output of the previous layer by a factor of two, and the final layer which creates the segmentation output.

Before the input images are passed to the first segment, they are downsampled by going through a 2D-average-pooling operation in 4×4 regions by step size of one. In the first segment, its CNN has 16 hidden layers. In each hidden layer, the input tensor gets padded by using reflection of the input boundary first. Then the padded layer comes to a convolutional layer which has an input channel of 64 and an output channel of 64. The convolutional layer in the first hidden layer is special. Its input channel is one. The activation function after the convolutional layer is Leaky ReLu whose negative slope is 0.01. At the end of each hidden layer, batch normalization is applied to the output from the Leaky ReLu activation function. The final output of the 16 hidden layers then comes to a deconvolutional layer, which uses a 2D transposed convolution operator.

The output of the first segment is then combined with a downsampled original input which is processed by a 2D-average-pooling operation in 2×2 regions by step of one to become the input to the second segment. In the second segment, its CNN has 2 hidden layers. In each hidden layer, the input tensor gets padded as the first segment. The convolutional layer in the first hidden layer has an input channel of 64, and an output channel of 72; while, the one in the second hidden layer has an input channel of 72 and an output channel of 72. The activation function, batch normalization in each hidden layer, and the deconvolutional layer are the same as the first segment.

We add up the original input and the output of the second segment to be the input of the final segment. The structure of CNN in the third segment is the same as the second segment, except that the convolutional layer in the first layer has an input channel of 72, and an output channel of 80, and the input channel as well as the output channel of the convolutional layer in the second hidden layer are both of size 80. The output of the third segment, eventually, is passed to the last layer, which consists of a padding layer, a convolutional layer whose input channel is 80 and output channel is 2, one sigmoid activation function for binary segmentation, one ReLU activation function for the prediction of each pixel's distance to the nearest cell boundary, and a batch normalization.In general, the J-net has 23 layers.

Compared to the output from DeepWater, the output images from the J-net are blurry and incomplete. To remedy the quality of its output, we apply post-processing using the watershed algorithm. Because there are some holes in the output masks, we apply morphological opening to fill in the missing parts, before which consecutive thresholding and Gaussian blur are applied. After the opening operation, we apply the watershed algorithm to get a clear segmentation.

4) Thresholding Segmentation: Histogram shape-based thresholding algorithm is applied in the Fluo-N2DL-HeLa and the PhC-C2DL-PSC dataset.

An observation of the images in Fluo-N2DL-HeLa reveals that the background color is the main component. By plotting a histogram of a gray image, we can find that the peak represents the intensity of the background. With such intensity, we can conduct thresholding to separate the foreground and background easily. After the separation, a morphological opening operation is utilized to remove noises and to fill small holes inside cell masks.

As for the PhC-C2DL-PSC dataset, a glance at the images indicates that the cell density is high and the gaps between cells are narrow. In case of overlapping masks, a morphological opening operation with a kernel size of two is applied to shrink the size of cells first. Afterward, a Gaussian filter is used to remove noise, to blur edges and to reduce contrast. With histogram utilized again, the peak is found, and the foreground and background can be separated by the thresholding algorithm.

B. Cell Detection

In both our application of neural networks and histogram-based thresholding for the segmentation of cells in each dataset, we obtain a binary mask image with light regions representing the cell body and dark regions representing the background. We then apply a border following algorithm to extract the topological structure of each cell [23]. The topological structure is represented by a set of coordinate points which describe the cell's contour as a collection of connected 1-pixel components.

To prevent false detection of noisy regions or extraneous particles (particularly for the PhC-C2DL-PSC dataset), we compute the area bounded by each contour and reject the contour as a cell candidate if its area falls below a certain threshold t_c, which we determine based on visual inspection of the average cell size present in each dataset. The remaining contours are considered cells and passed into the matching algorithm which is considered further below.

C. Cell Tracking

In order to track cells across the spatial-temporal context of each sequence of images, we devise an algorithm which is similar to detection association frameworks yet incorporates elements of model evolution to exploit the temporal context. A hybrid framework is better able to address some of the shortcomings of model evolution in the detection of cell behaviours such as mitosis [24]. In our proposed methodology, cell segmentation is performed independently for each frame and used to update a model of the universe of cells existing from frame to frame. However, to prevent errors from propagating across consecutive cell states (whether as a result from poor segmentation or matching error) the model is only weakly linked in the temporal dimension. After each frame is segmented into cell contours, the model of cell states resulting from the previous frame is updated to incorporate information in the current frame only where a correspondence between cells is successfully found. Where no such correspondence is found, either as a parent-child relationship, or a predecessor-successor relationship, the cell is considered new and is included in collection of cell states associated with the current frame. This approach improves the robustness of model evolution while being able to incorporate detection of certain cell behaviours including mitosis.

Our proposed matching algorithm is as follows. We calculate the Euclidean distances of centroids of each cell which has been registered in the cell state for the $t-1$ frame to each cell in the t frame, and select the two cells in the t frame with the smallest distance as candidates for matching. A match is then determined by generating the areas bounded by the candidate cell contours and calculating their ratio to the cell in the previous frame. If both the closest and second candidates' areas ratios lie within a range a_{min} to a_{max} and their distance is smaller than some threshold d_t, then mitosis is detected. We update the candidate cells to indicate they are child cells, and the previous frame's cell to indicate it is a parent cell.

If mitosis is not detected, we instead test for a predecessor-successor relationship between the cell in the $t-1$ frame and its closest cell in the t frame. If the ratio of areas between the successor candidate and the previous frame cell lies within a range b_{min} to b_{max}, and the minimum distance threshold d_t is satisfied, the predecessor is associated with its successor by retrieving cell's state in the previous frame and updating it with new position information in the current frame. By updating cell states in this manner, cells are linked in the temporal dimension and a history of their topological features and motion can be constructed from frame to frame.

Any cells which fail to be associated either in a parent-daughter cell relationship (mitosis) or a predecessor-successor relationship are registered as new cells in the current frame. We adopt this approach to preserve the integrity of the cell base in the current frame, and to prevent the propagation of matching errors from previous frames to later frames. Accordingly, each frame can be viewed as a weakly connected cell state between t, $t-1$ and $t+1$, with transitional information captured by propagating information from the current frame to the previous frame, and the next frame to the current frame.

Dewan et al. adopt a similar approach to mitosis detection by using reverse template matching [15]. However, a significant downside to this technique in its most basic form is the loss of real-time processing as disconnected cell trajectories must be traversed in reverse temporal order to detect mitosis and recover these connections. By limiting our temporal window to the next frame only, our approach achieves early trajectory recovery in real time. We note, however, the importance of good segmentation results for cell tracking (in contrast to cell detection) in our algorithm as with other frameworks based on detection association.

D. Cell Trajectory and Motion

In the above methodology, each input image produces a collection of cell objects that are either associated with the cell objects in the previous frame via a parent-child relationship or as predecessor and successor. We save these cell states and their frame associations as each input image is processed and matching is performed. To calculate cell trajectory and motion characteristics, each cell object is labelled with a cell ID and retains a history of its previous centroid positions, as well as its current centroid, contour, area and mitosis status (either as a parent of a cell in the next frame, or a child of a cell in the previous frame). In the matching phase, a successful match produces an update to the cell object at the earlier time step with its new position, area and contour information. Accordingly, each cell's trajectory can then be reconstructed using the cell's position history. This update also triggers a recalculation of the cell's speed, net distance travelled, total distance travelled, and confinement ratio. As each frame-by-frame cell state is saved, we allow for retrieval, annotation and display of this information by the user after the image sequence has been processed.

IV. Experimental Setup

The image data we used is taken from the international Cell Tracking Challenge [25]. We process three data sets: DIC-C2DH-HeLa, Fluo-N2DL-HeLa and PhC-C2DL-PSC. Each of them contains four sequences of time-lapse microscopy from different biological experiments. We conduct experiments on Google Colab Pro with HIGH RAM runtime mode, and the GPU we used is Python 3 Google Compute Engine Backend. Significant Python packages are: tensorflow, keras, PyTorch, Numpy, OpenCV, SciPy, scikit-image and tqdm.

We get the code of J-net from [20], and with code modifications and after 3,000 epochs training on Google Colab, we get our J-net model for segmentation.

Qualitative evaluation can make the results more intuitive, while quantitative evaluation can make the results more precise and convinced. Therefore, we decide to evaluate our proposed method in both of these two ways.

A. Evaluation of Segmentation

The result processed by our cell segmentation methods are illustrated in Fig. 1. In this part, we use three different ways to deal with three different data sets in accordance with their nature to achieve a better performance. Fig. 1(a) depicts the frame $t001$ of sequence 1 in data set DIC-C2DH-HeLa. The cells in the frame are large and close to the other cells. Fig. 1(b) shows the result of a J-net mask processed by the watershed transformation and Fig. 1(c) reveals the cell contours. From these three images we can see that there is some under-segmentation in the results. Fig. 1(d) shows the result mask of DeepWater network and Fig. 1(e) reveals the contours of the cells. We obtain a clear view of correctly separated biological cells in these two images. Fig. 1(f) shows frame $t000$ of sequence 1 in dataset Fluo-N2DL-HeLa. This sequence has low contrast. Fig. 1(g) shows the result of our histogram-based thresholding method and Fig. 1(h) shows the contours of the separated cells. From these two images, we can clearly see that the biological cells are divided correctly. Fig. 1(i) shows the original frame $t000$ of sequence 1 in data set PhC-C2DL-PSC. Cells in this sequence are populated intensively. Fig. 1(j) indicates the result of histogram-based thresholding and Fig. 1(k) reveals the contours of the separated cells. Fig. 1(l) presents the result mask of the DeepWater network and Fig. 1(m) depicts the contours of the cells. Both of these methods achieve a good result.

We now present a quantitative evaluation of the proposed method. We evaluate the method using the schemes from the online version of the Cell Tracking Challenge [25]. The segmentation evaluation method is defined as Number Ratio NR:

$$NR(A,B) = \frac{NCD(A \cap B)}{NCD(A \cup B)} \tag{3}$$

where $NCD(A)$ is the number of connected domain of the set A. A is the ground truth label, and B is its corresponding segmentation. It is a statistic used for gauging the number ratio and similarity of sample sets. If $NR(A,B)$ is larger than 1,

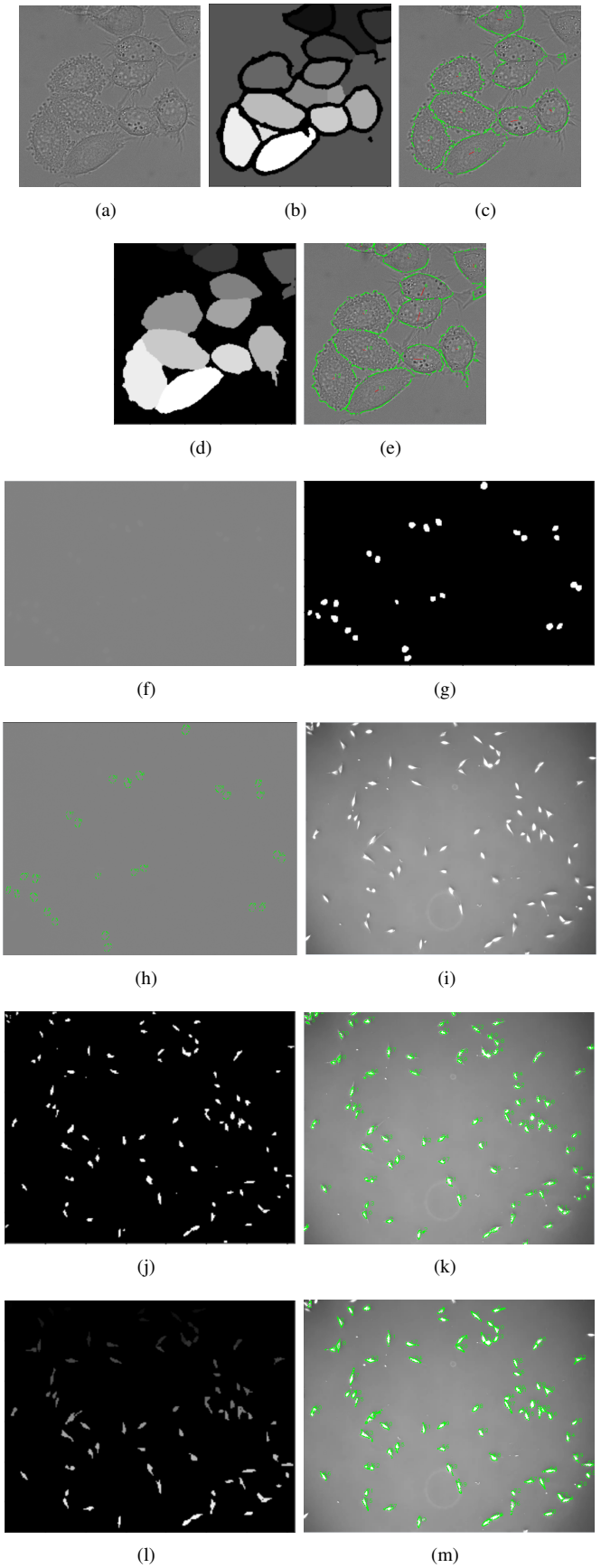

Fig. 1. Illustration of the segmentation of the cells used in the proposed method. (a) Frame $t001$ of sequence 1 in DIC-C2DH-HeLa. (b) J-net mask processed by Watershed algorithm. (c) Contours of (b). (d) Masks from the DeepWater network. (e) Contours of (d). (f) Frame $t000$ of sequence 1 in Fluo-N2DL-HeLa. (g)Thresholded cells of (f). (h) Contours of (g). (i) Frame $t000$ of sequence 1 in PhC-C2DL-PSC. (j) Thresholded cells of (i). (k) Contours of (j). (l) Masks from DeepWater network. (m) Contours of (l).

it means there are some over-segmentation. If $NR(A, B)$ is smaller than 1, it means some of the cells are not detected. The smaller difference between $NR(A, B)$ and 1, the larger the similarity between the two sets.

Another segmentation evaluation method is defined as the Jaccard index:

$$J(A, B) = \left| \frac{A \cap B}{A \cup B} \right| \tag{4}$$

where A is the ground truth label, and B is its corresponding segmentation. The Jaccard index compares the area of two sets and is a statistic used for gauging similarity of sample sets. Larger Jaccard index means higher similarity.

For a quantitative evaluation of segmentation, we compute the Number Ratio and Jaccard index between our result masks and the ground truth masks available on the Cell Tracking Challenge. Ground truth masks were available for Sequence 1 and 2 of each dataset. Table I shows the details of datasets and sequences we chose for evaluation, the methods assessed, and the mean value as well as the minimum of Number Ratio, while, Table II provides the same for the Jaccard index. Both the DeepWater network and the J-net work well on the DIC-C2DH-HeLa set, but DeepWater outperforms as the mean value of Jaccard index is 0.84 while J-net is 0.74. Our thresholding algorithm achieves a 0.5335 NR in sequence 1 in Fluo-N2DL-HeLa, and 0.7285 in sequence 2. The Jaccard index of thresholding in both sequences 1 and 2 are also not ideal. As for the third data set PhC-C2DL-PSC, the performance of DeepWater network is better than thresholding, even though the number ratio of former exceeds 1, which means there are some overlapped cells not segmented well. The Jaccard index table reveals the same judgement as well.

B. Evaluation of Tracking

To demonstrate the qualitative evaluation of our proposed tracking method, we randomly choose three result images from each of the three data sets and look backward to check their trajectories. For the DIC-C2DH-HeLa data set, we select frame $t029$ from its sequence 2. Fig. 2(a) shows the frame $t000$ and Fig. 2(b)-(l) present the enlarged views of the top left part of images from frame $t019$ to $t029$. The red lines represent the trajectories of the cells, and the blue contour means that the cell is in its mitosis period. From these figures we can see that the trajectories of the cells are correct. Fig. 3 and Fig. 4 reveal the trajectories of the cells from Fluo-N2DL-HeLa and PhC-C2DL-PSC respectively.

V. Results and Discussion

Statistical and visual results are listed in the evaluation part(Fig. 1, Fig. 2, Fig. 3,Fig. 4 and Table I). We now discuss the performance of each algorithm and the outcomes of the experiment. Firstly, our methodology is successful in detecting cells in each dataset and drawing a bounding box around each of them. To retain cell boundary information, we decided to display such bounding boxes directly as cell contours rather than fit a polygon. We also draw the trajectory of each cell and print the real-time count of the cells detected in each image of the sequence. Fig. 5 shows the detailed motion characteristics of cell 6 from frame $t006$ in sequence 3 of data set DIC-C2DH-HeLa. From this figure we can see that the detected cells have green contours, and the trajectory of each cell is marked as a red line, and the real-time count of the cells detected is 13. From Fig.5, it could be seen that there are no cells surrounded by blue contours and this indicates there are no cells undergoing mitosis in this frame. The zero count of diving cells also support this. Fig. 5 shows that cells that are dividing have blue contours and their real-time count is 1. After that, we compute some properties for a selected cell up to the current time point, including speed, total distance, net distance, and confinement ratio. Fig. 5 reveals that the speed of the cell with ID 6 is 1.41 pixel/frame. And total distance and net distance travelled up to that time point are 7.26 pixels and 4.47 pixels respectively. As for the confinement ratio of the chosen cell's motion, it is 1.62, indicating that the cell has a moderate degree of drift in the region near to its starting position. Fig. 6 presents the results of dataset Fluo-N2DL-HeLa and Fig. 7 shows the outcome of PhC-C2DL-PSC.

We use two methods, DeepWater network and J-net, to deal with the images in dataset DIC-C2DH-HeLa. Fig. 8(a) presents the original frame $t001$ from the first sequence. Fig. 8(b) and Fig. 8(c) display the output results of the highlighted portion in Fig. 8(a) processed by J-net and DeepWater respectively. Fig. 8(d) reveals frame $t084$ from the first sequence, and Fig. 8(e) and Fig. 8(f) show the image of the highlighted

TABLE I
NUMBER RATIO EVALUATION OF CELL SEGMENTATION

Data	method	NR(A,B) mean	NR(A,B) min
DIC-C2DH-HeLa Seq.1	DeepWater	1.536	0.3056
DIC-C2DH-HeLa Seq.1	Jnet	1.024	0.109
DIC-C2DH-HeLa Seq.2	DeepWater	1.2267	0.2308
DIC-C2DH-HeLa Seq.2	Jnet	1.606	0.3143
Fluo-N2DL-HeLa Seq.1	threshold	0.5335	0.4240
Fluo-N2DL-HeLa Seq.2	threshold	0.7285	0.6407
PhC-C2DL-PSC Seq.1	DeepWater	1.2455	1.1000
PhC-C2DL-PSC Seq.1	thresholding	0.8384	0.7593
PhC-C2DL-PSC Seq.2	DeepWater	1.2421	0.9776
PhC-C2DL-PSC Seq.2	thresholding	0.8353	0.7234

TABLE II
JACCARD INDEX EVALUATION OF CELL SEGMENTATION

Data	method	J(A,B) mean	J(A,B) min
DIC-C2DH-HeLa Seq.1	DeepWater	0.8409	0.6241
DIC-C2DH-HeLa Seq.1	Jnet	0.7426	0.1352
DIC-C2DH-HeLa Seq.2	DeepWater	0.8460	0.5899
DIC-C2DH-HeLa Seq.2	Jnet	0.7891	0.5684
Fluo-N2DL-HeLa Seq.1	threshold	0.3232	0.2546
Fluo-N2DL-HeLa Seq.2	threshold	0.4900	0.4414
PhC-C2DL-PSC Seq.1	DeepWater	0.6993	0.6364
PhC-C2DL-PSC Seq.1	thresholding	0.4042	0.3493
PhC-C2DL-PSC Seq.2	DeepWater	0.7092	0.6767
PhC-C2DL-PSC Seq.2	thresholding	0.4131	0.3803

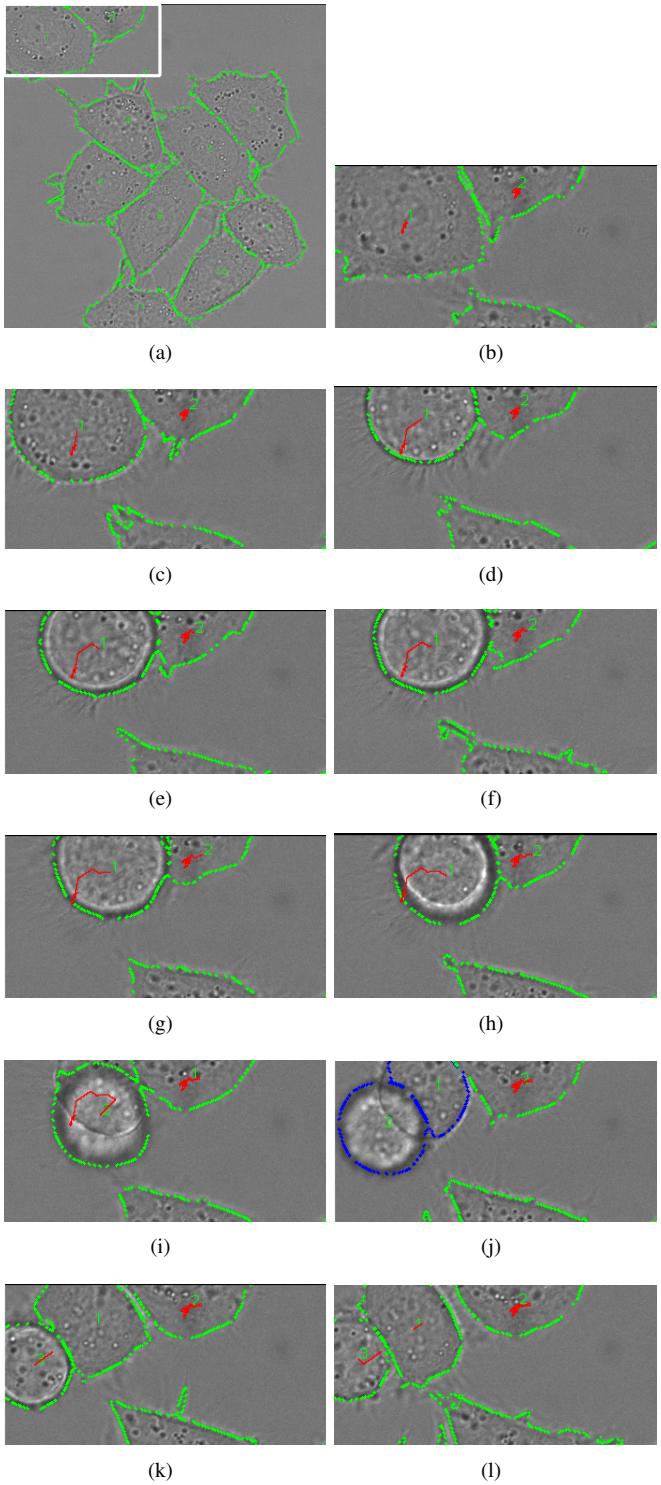

Fig. 2. Exhibition of the tracking of the cells used the proposed method. (a)Frame $t000$ of sequence 2 in DIC-C2DH-HeLa. (b)-(l)The enlarged views of the top left part of images from frame $t019$ to $t029$.

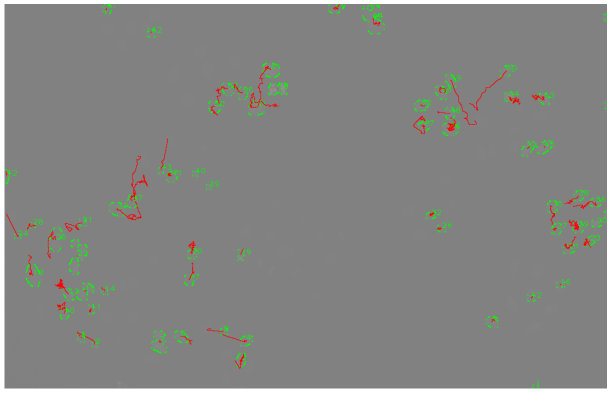

Fig. 3. Demonstration of the tracking of the cells from Fluo-N2DL-HeLa dataset using the proposed method.

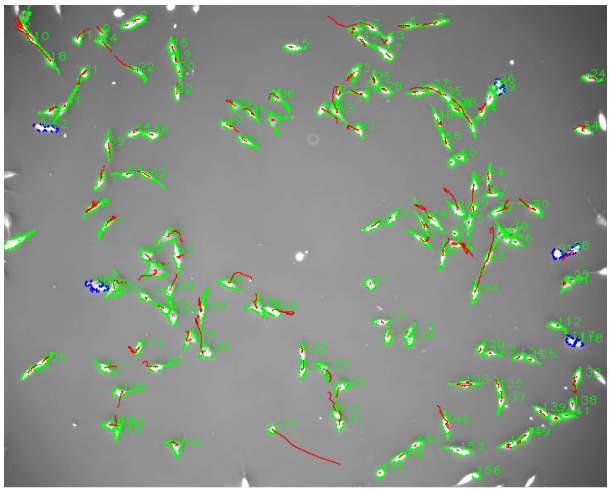

Fig. 4. Illustration of the tracking of the cells from PhC-C2DL-PSC dataset using the proposed method.

portion of Fig. 8(d) processed by J-net and DeepWater respectively. From these figures we can see that the performance of DeepWater is better than J-net. Moreover, we reach the same conclusion from the Number Ratio and Jaccard index. As for the dataset Fluo-N2DL-HeLa, we obtain a number ratio about 0.6, and the mean Jaccard index of two sequences is about 0.4. Though it is not as high as the others, it means some of the cells can be detected correctly by our thresholding algorithm. From Fig. 4, Table I and Table II, we can see that our proposed method also works well for the PhC-C2DL-PSC dataset.

VI. Conclusion

This paper presents our proposed method for automatically tracking and analyzing biological cell behaviors in a continuous time series. This method is able to solve some of the problems caused by high density of cells, low contrast, unclear contours, irregular motion and mitosis. Our method implements cell segmentation by applying the both deep learning frameworks of J-net and DeepWater, and our own thresholding algorithm developed using non-learning based computer vision techniques. In the case of deep learning, we improves the

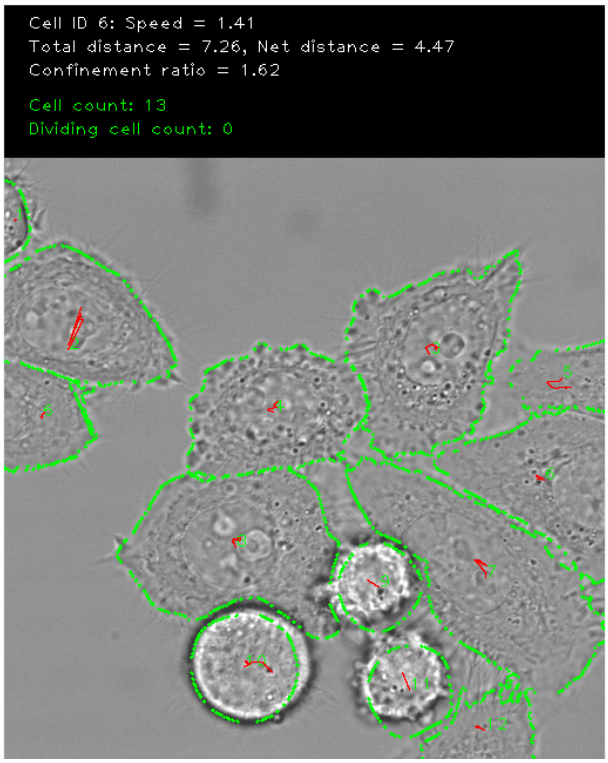

Fig. 5. Result of cell 6 from frame t006 in sequence 3 of dataset DIC-C2DH-HeLa.

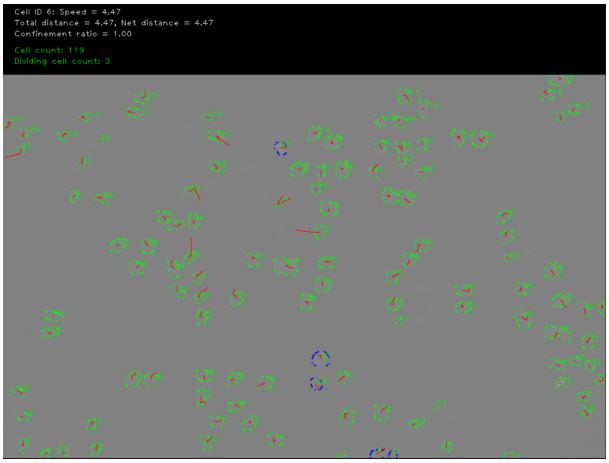

Fig. 6. Result of cell 6 from frame t005 in sequence 2 of dataset Fluo-N2DL-HeLa.

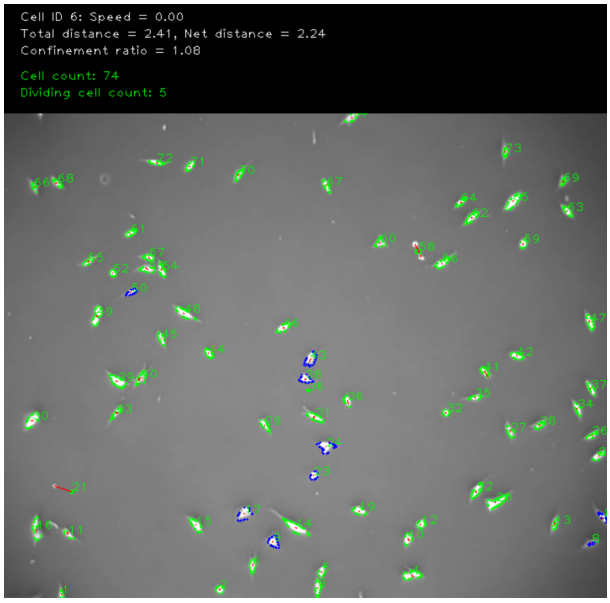

Fig. 7. Result of cell 6 from frame t006 in sequence 4 of dataset PhC-C2DL-PSC.

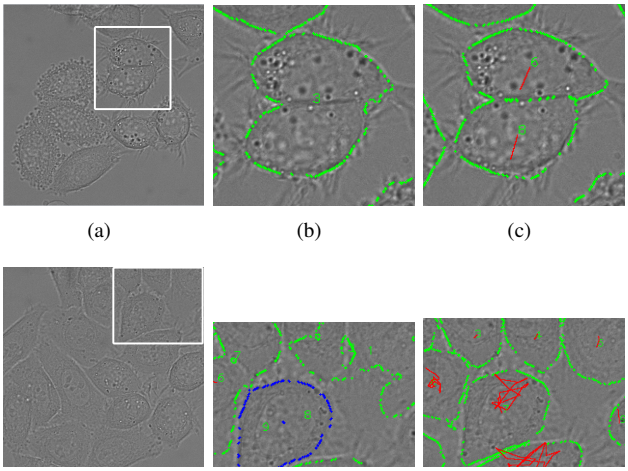

Fig. 8. Illustration of result. (a) Input image. (b) Image of the highlighted portion of (a) processed by J-net. (c) Image of the highlighted portion of (a) processed by DeepWater. (d) Input image. (e) Image of the highlighted portion of (d) processed by J-net. (f) Image of the highlighted portion of (d) processed by DeepWater.

segmentation results by applying post-processing using various morphological methods in computer vision. Since these deep learning frameworks have good adaptability, they could be generalized on new data to provide valid models for cells with different characteristics.

In the cell tracking phase, we use distance and area in cell topology as features to identify cell motion as well as mitosis, and record a cell's trajectory based on the identifications and associations of cell behavior from backward to forward in sequential frames. This method has a good performance for tracking cells even during cell division. However, we identified that our thresholding algorithm could be refined further to address segmentation issues relating to touching cell boundaries and the tracking of fast moving cells. Although we propagate mitosis information to the previous frame, our algorithm could also be extended to trace cell division back to an earlier time point in case that mitosis spans multiple consecutive frames.

The evaluated results have demonstrated that the methods proposed in our work achieve reasonably strong performance and stability in detecting, tracking and analysing biological cells. Hence, these methods could be applied in real applica-

tions like drug discovery and cancer research.

VII. Contribution of Group Members

State each group member's contribution in brief. In at most 3 lines per member, describe the component(s) each group member contributed to.

- **Minrui Lu:** Training of J-net model, integration of J-net and DeepWater models, final program integration, methodology of J-net and DeepWater parts in this paper, proofreading of this paper, demo preparation.
- **Shuang Liang:** Consultation with Deep Learning researchers and cell segmentation with approaches in OpenCV; abstract, literature review, and conclusion parts of this report.
- **Xiaocong Chen:** Perform the experiments with approaches in OpenCV; write introduction, experimental setup, and results and discussion parts of this report.
- **Yunfan Wang:** Design and realize codes for the simple threshold, cell tracking and result presenting methods. Refine and integrate the final program version.
- **Con Tieu-Vinh:** Program design and development; methodology section and proofreading of this paper; demo preparation.

Acknowledgement

We would like to express our great appreciation to Dr. Filip Lux for his generous sharing of his DeepWater Model and code, and to Tomas Sixta for sharing his J-net code on Github.

References

[1] E. Meijering, O. Dzyubachyk and I. Smal, "Methods for cell and particle tracking," Methods in Enzymology, vol. 504, no. 9, pp. 183-200, February 2012.

[2] K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S.L. Schmid, G. Danuser, "Robust single-particle tracking in live-cell time-lapse sequences," Nature Methods, vol. 5, no.8, pp. 695-702, 2008.

[3] R. Chatterjee, M. Ghosh, A. S. Chowdhury, N. Ray, "Cell tracking in microscopic video using matching and linking of bipartite graphs," Computer Methods and Programs in Biomedicine, vol. 112, no.3, pp. 422-431, December 2013.

[4] K. Li, E. Miller, L. Weiss, P. Campbell, T. Kanade, "Online tracking of migrating and proliferating cells imaged with phase-contrast microscopy," Conference on Computer Vision and Pattern Recognition Workshop, pp. 65-72, 2006.

[5] N.N. Kachouie, P. Fieguth, J. Ramunas, E. Jervis, "Probabilistic model-based cell tracking," International Journal of Biomedical Imaging, pp. 1–10, 2006.

[6] E. Meijering, O. Dzyubachyk, I. Smal, W. A. van Cappellen, "Tracking in cell and developmental biology," Seminars in Cell and Developmental Biology, vol. 20, no. 8, pp. 894-902, October 2009.

[7] E. Meijering, "Cell Segmentation: 50 Years Down the Road [Life Sciences]," IEEE Signal Processing Magazine, vol. 29, no. 5, pp. 140-145, Sept. 2012.

[8] E. Moen et al, "Deep learning for cellular image analysis," Nature Methods, vol. 16, no. 12, pp. 1233-1246, December 2019.

[9] O. Ronneberger, P. Fischer, T. Brox, "U-net: Convolutional networks for biomedical image segmentation," Med Image Comput Comput Assist Interv, pp. 234–241, 2015.

[10] DA. Van Valen, T. Kudo, K.M. Lane, D.N. Macklin, N.T. Quach, M.M. DeFelice, I. Maayan, Y. Tanouchi, E.A. Ashley, M.W. Covert, "Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments," PLoS Comput Biol, vol. 12, no. 11, p.e1005177, 2016.

[11] S. K. Nath, F. Bunyak, K. Palaniappan, "Robust tracking of migrating cells using four-color level set segmentation," International Conference on Advanced Concepts for Intelligent Vision Systems. Springer, Berlin, Heidelberg, pp. 920-932, 2006.

[12] X. W. Chen, X. B. Zhou and S. T. C. Wong, "Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy," IEEE Transactions on Biomedical Engineering, vol. 53, no. 4, pp. 762-766, April 2006.

[13] C. Wählby, I.M. Sintorn, F. Erlandsson, G. Borgefors & E. Bengtsson, "Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections," Journal Of Microscopy, vol. 215, pp. 67–76, 2004.

[14] E. Bengtsson, C. Wahlby, J. Lindblad, "Robust cell image segmentation methods," Pattern Recognition and Image Analysis C/c of Raspoznavaniye Obrazov i Analiz Izobrazhenii., vol.14, no. 2, pp. 157-167, 2004.

[15] M. A. A. Dewan, M. O. Ahmad, M. N. S. Swamy, "Tracking biological cells in time-lapse microscopy: An adaptive technique combining motion and topological features," IEEE transactions on biomedical engineering, vol. 58, no. 6, pp.1637-1647, 2011.

[16] O. Debeir, P. Van Ham, R. Kiss, et al, "Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes," IEEE transactions on medical imaging, vol. 24, no. 6, pp. 697-711, 2005.

[17] N. Ray, S. T. Acton, K. Ley, "Tracking leukocytes in vivo with shape and size constrained active contours," IEEE transactions on medical imaging, vol. 21, no. 10, pp. 1222-1235, 2002.

[18] J. Yan, X. Zhou, Q. Yang, et al, "An effective system for optical microscopy cell image segmentation, tracking and cell phase identification," 2006 International Conference on Image Processing. IEEE, pp. 1917-1920, 2006.

[19] L. Filip and P. Matula. "Cell Segmentation by Combining Marker-Controlled Watershed and Deep Learning," arXiv.org, 2020.

[20] T. Sixta, "J-net," Github repository, https://github.com/tsixta/J-net, 2018.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241, 2015.

[22] S. Evan, J. Long, and T. Darrell. "Fully Convolutional Networks for Semantic Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence 39, no. 4, pp. 640–51, 2017.

[23] S. Satoshi and A. Keiichi, "Topological structural analysis of digitized binary images by border following," Computer Vision, Graphics, and Image Processing, vol. 30, no. 1, pp. 32-46, 1985.

[24] J. Chen, M. Alber and D. Chen, "A hybrid approach for segmentation and tracking of myxococcus xanthus swarms," IEEE Trans Med Imaging, vol. 35, no. 9, pp. 2074-2084, 2016.

[25] V. Ulman et al, "An objective comparison of cell-tracking algorithms," Nature Methods, vol. 14, no. 2, pp.1141-1152, December 2017.