Tracking Biological Cells Division and Migration

*COMP9517 20T2 Project

***Abstract*—This paper aims to design an algorithm to collect and analyze cell division and migration in real time by computer vision methods. In terms of the three steps, these are Cell segmentation, tracking cell trajectories and division over time and analyzing a series of parameters of cells movement. For cell segmentation and tracking, we use advanced U-Net followed by a watershed segmentation algorithm and dissimilarity measure which combines cell movement features and cell topological features, respectively. In terms of movement analysis, we apply ROI method which we could select a cell at any time point and then analyze the motion of the selected cell. All the target datasets come from three different types of image sequences and we get high accuracy in detecting and tracking cells finally.**

***Index Terms*—Cells, Mitosis, Trajectory, Unet, Movement**

I. Introduction

Cell division and migration are two essential processes in the cellular cycle. Although optical microscopy is able to visualise these processes efficiently, resulting information including image sequences and videos are hard to be analysed by human vision because of enormous quantities and indistinguishable details. Therefore, it is necessary to apply certain technologies of computer vision to cell biology so that it could effective to automate detection, tracking and analysis cells.

In terms of the goal of this project, it aims to utilise suitable computer vision and machine learning models to segment cell boundaries and tracking cell movements over time and provide certain information about the procession of cell division and migration.

The datasets used for this project are three different types of cells sequences. For each type of cells, it contains four sequences of different timelines and a part of masks for certain time points. Therefore, we could use these masks and corresponding original cell images to train our model and predict the rest of the images so that we could efficiently detect and track cells throughout the timeline. Additionally, among these three different types of cell sequences, cells of the first dataset named “DIC-C2DH-HeLa” are relative hard to detect because of relative blurred boundaries and overlapping.

More specifically, Figure 1 shows the details of three random cases in each dataset. Subfigures (a), (d), (g) are input images of three datasets respectively and subfigures (b), (e), (h) are corresponding intensity histograms for these three images. Besides, subfigures (c), (f) and (i) are the certain rows of these three images so that it could reflects the internal differences from cells and the background. As the Figure 1 showed, in terms of the subfigure (d), although the pixels of cells and its background are close obviously, the distribution

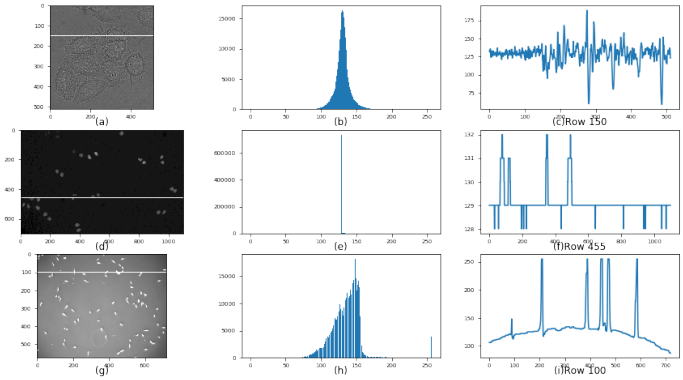


Fig. 1. Cases among Three Dataset. (a) One grayscale image of first image sequences dataset ’DIC-C2DH-HeLa’. (b) Intensity histogram of (a). (c) Horizontal distribution of intensity for (a) at row 150. (d) One grayscale image of second image sequences dataset ’Fluo-N2DL-HeLa’. (e) Intensity histogram of (d). (f) Horizontal distribution of intensity for (d) at row 455. (g) One grayscale image of second image sequences dataset ’PhC-C2DL-PSC’. (h) Intensity histogram of (g). (i) Horizontal distribution of intensity for (g) at row 100.

of these two parts are easy to be detected, which means there exists obvious threshold (approximate pixel 130) between cells and background. For the details of row 455 showed in subfigure (f), obvious peaks and valleys could be seen which represent the cells (bright areas) and the background (dark areas) respectively. For the second dataset and its random case, subplot (g), there also exists the obvious threshold between cells and background which cells are around 255 pixel and the range of background are between 50 and 200. According to the subfigure (i), there are also obvious peaks and valleys and compared with the subfigure (d), although subfigure (c) has more noises and larger range of background, it is also easy to detect cells. Finally, compared with the last two datesets, it is harder to segment cells in the first dataset. In terms of subfigure (a), there are various noises in the background and all of the cells are hollow which means the edge of cells is the most obvious determining factor to segment cells from the background. However, the discontinuous boundaries and a large number of noises and overlapping would greatly reduce the distinction of cells. In the subfigure (i), there are not obvious peaks and valleys at the row 150 of subfigure (a) wihch means apply thresholding method directly is ineffective and would generate various false segmentation. Besides, typical method like Watershed also has worse performance due to high noise sensitivity

which would generate over-segmentation. Therefore, for the image sequences of first dataset, there need additional special treatments to solve these problems such as machine learning method. More details are showed is the Method chapter.

For the processes of this project, there are three connecting steps. For the first task, we need to apply our model to detect and track cells of all the dataset and print the real-time quantity of cells for each time points. The second task aims to identify the process of cell division. For the last task, we need to analyze and obtain certain parameter of cell motive include speed, distance etc. All of these three tasks are built on the dataset with sequences of three different type of cells.

II. Literature Review

Considerable research effort on cell segmentation and cell tracking has been made in the past few years. In order to better understand the techniques and their background, we review a lot of relevant techniques in literature.

For cells segmentation, deep learning strategies are able to reduce the errors caused by merged objects, and provide tighter and smoother segmentation boundaries than the boundaries estimated by global Otsu thresholding [1] [2] [3] and declumping [1]. Thresholding [1] [2] [3] is only suitable to well separated and background without severe noise. Besides, U-Net [1] [3] achieves a more precisely F1 score in nucleus segmentation and preforms better at correct splitting of adjacent nuclei than DeepCell [1] [3], the advanced and basic CellProfiler pipelines. Compared to DeepCell which has 2.5 million parameters, U-Net architecture has 7.7 million parameters, but U-Net runs faster than DeepCell when using GPU acceleration. However, compared to classical methods deep learning requires more calculation and annotation time. Predetermined cell intensity profiles (templates) [2] are complex methods for image segmentation. With this template fitting to the image data, consistent cell shape can be shown, but if cell morphology changes a lot, cell segmentation may fail. Another approach is the so-called watershed transform [2] [3] which combines well-designed pre- and post-processing strategies. Although it has been used to cell segmentation in microscopy, but it is sensitive to noise and prone to produce over-segmentation. In the contrast, model-evolution approaches are easy to produce under-segmentation of the images, and usually need post-processing steps. Object-detection-based deep learning methods [3] have successful segment instance when adapted to cellular data [4] [5] [6]. Besides, a discriminative loss function match pixels to vectors, in this case, deep learning models could improve instance segmentation performance accurately even when objects overlap. Deformable models [2] which define as parametric contours or the zero-level level-set function become popular for cell segmentation. Compared to parametric contours, the zero-level level-set function perform well in capturing topological changes.

For cells tracking, the algorithms can roughly be divided into tracking by detection [7] [8], tracking by model evaluation [7] [8], and tracking by filtering [8]. Although tracking by detection approaches [7] [8] outperformed tracking by contour evolution approaches [7] [8] and the nearest neighbors approaches [2] [7] [8](associating the cell to the spatially nearest cell in the next frame is used to obtain cell trajectories) in most scenarios, compared to tracking by detection methods, tracking by contour evolution methods performs better if the population density of cells is too high or cells move rapidly. The nature of Tracking by filtering approaches [8] is probabilistic and the methods are commonly used in the tracking of virus particles, but because of the low signal-to-noise ratio, it is a hard problem. In addition, probabilistic tracking methods(Bayesian estimation) can infer all information(such as cells position, shape) from noisy observations. In linear-programming approaches, multiple behaviors such as disappearance, splitting, and merging can be solved because the center of object, intensity, and morphology are considered to link objects between frames [3]. The method based on Gaussian mixture models [9] [2] is proposed to detect and track overlapping cell nuclei of large amount mitosis. 2D Gaussian mixture is used to fit the model to the data and two parameters(an average nucleus diameter $\bar{d}_{nuc}$ and intensity $\bar{w}_{nuc}$) are used to ease the convergence. GMM models with two and three components are used to features extraction and achieve a high throughput. The result of overlapping detection and cells division in Cellprofiler after a seeded Watershed segmentation perform not well. Compared to the manual annotation, the distributions of angles obtained automatically were very similar. However, possible bias may be caused by the statistical problems.

III. Methods

A. Segmentation Based Detection

The detection of cells follows the procedure that two U-Nets predict masks and markers for the cells in an image, followed by a marker controlled watershed segmentation algorithm which uses masks and markers as sure background and sure foreground. [10]

The prediction part is supervised. Before training the model, we must do data augmentation and normalisation. Once the predicted masks and markers are obtained, the noise needs to be eliminated as much as possible. Then, the processed masks and markers can be used to do segmentation.

1) Data Preprocessing:

a) Mask Error Correction: Noticed the shortage and inaccuracy of provided masks, we decided to use the original dataset from CTC. However, there are still "missing cells" in masks of DIC-C2DH-HeLa. So, we need to manually fill the vacancies.

b) Mask Erosion: The mask U-Net is designed to predict binary images, which is to be segmented. It means we want small "cracks" among cells, but mask edges especially of DIC-C2DH-HeLa largely contact with each other and hard to be segmented once they are binarized. So, all the contours must slightly erode.

Meanwhile, the masks also needs to erode for preparing the background truth for marker training. The difference is the erosion will be heavier.

the erosion structuring element with a diameter defined as:

$$d_{SE} = (1 - k) \cdot e_{min}$$

where k is a user-set coefficient and e_{min} is the length of min edge of min area rectangular of each contour. Following this formula, we are allowed to prepare background truth for both mask and marker training by defining different value of k.

c) Data Augmentation: In order to improve the robustness of U-Net model, we introduce new patterns into training and test dataset by generating scaled, rotated and flipped samples from the original ones. All images will rotate by a random degree from (-180°,180°), scale randomly by range of (0.6,1.4) and horizontally or vertically flip. The corresponding masks will do the same thing. Since the angle of microscopy view does not change, there is no need to do affine transformation.

d) Data Normalisation: The optimal choice of the normalisation function largely depends on the datasets. Different sequence acquisition methods and cell appearances produce different problems. For example, uneven illumination, low contrast and fluorescence.

In this case, we resize all the images to 512×512. Use histogram equalisation for DIC-C2DH-HeLa. Apply contrast stretching on Fluo-N2DL-HeLa, then set pixels equal or darker than the mode to 0 and finally apply histogram equalisation. Apply median scaling on PhC-C2DL-PSC.

Finally, normalise each image to [-0.5,0.5].

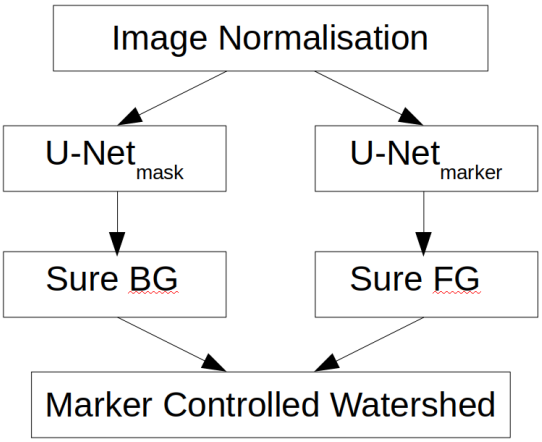


Fig. 2. Segmentation model Structure

2) U-Net Training: We use one network $U - Net_{mask}$ to predict cell mask pixels and the other network $U - Net_{marker}$ to predict the cell marker pixels. The raw output of prediction is the probabilities of existence of mask or marker for each pixel.

a) U-Net Structure: As Figure3 shows, the model architecture we implied is slightly different with the original U-Net [11]. We insert a batch normalisation layer between each convolutional layer and its activation function. Each network consists 11 U-Net blocks, each of which has 2 convolutional layers with kernel of 3×3 and ReLU activation function. There are also skip connections between blocks. The output layer is also a convolutional layer with kernel size of 1×1 and sigmoid activation function.

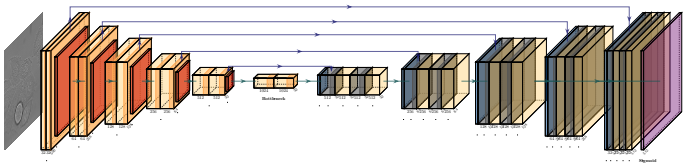


Fig. 3. U-Net Structure

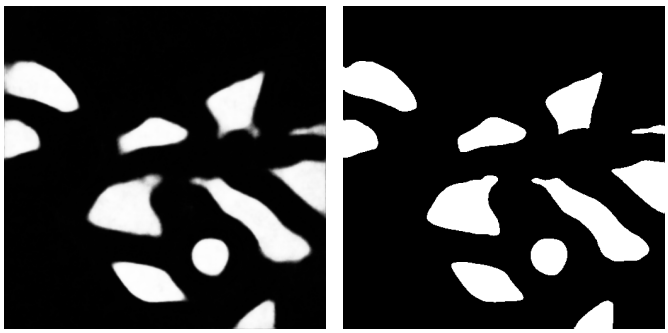
Though this network is a fully convolutional network and the size of image does not matter, we still **resize all images to 512×512 to train them together to get better generalisation performance.**

The loss function we applied is MSE loss function.

b) Training Procedure: We train each network for up to 200 epochs and save every model of higher accuracy on the test set. We calculate the accuracy by adding a round function at the output, counting pixels that are correctly predicted and dividing the count number by total pixel number.

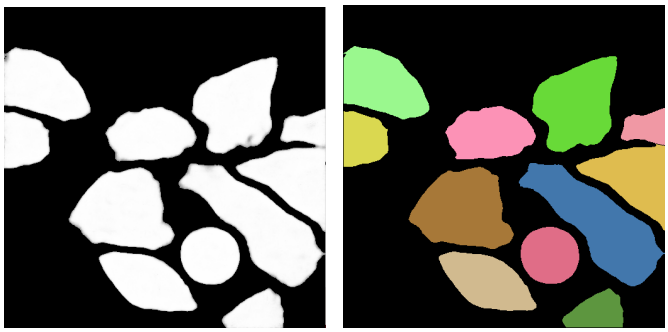
We also applied an Adam optimiser to find network parameters, whose initial learning rate is 3×10^{-4}.

3) Marker Function: After binarizing the predicted marker, we eliminate contours that are too small, whose radius is smaller than one tenth of the maximum radius.



(a) predicted marker (b) sure foreground

Fig. 4. Input and output of marker function



(a) Figure 4(b) corresponded mask (b) segmented mask

Fig. 5. Corresponding masks

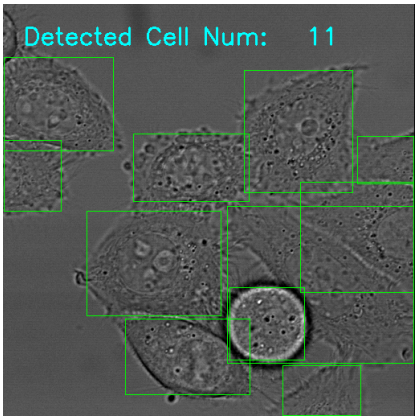


Fig. 6. original image with detected cells.

4) Marker Controlled Watershed: Now we do segmentation to predicted mask like Figure 5(a). The sure background is the threshold of mask.

B. Tracking & Division Detecting

In the process of cell tracking, it is not a simple task to achieve the multi-cell tracking of each frame of multi-cell sequence images with different motion characteristics. Due to cell migration, division, and the topology of cells changes over time, which makes it difficult for tracking work to find the appropriate relationship between cell characteristics [12].

Because a single feature is not enough to correctly match different units along a long image sequence. In view of this, we have combined cell movement features such as displacement and cell topology features such as deformation to improve the accuracy of matching.

According to the results of image segmentation, we get the X-axis coordinates C_x, Y-axis coordinates C_y, width C_w and height C_h of each cell in each frame of the image.

The $D_{displacement}$ is defined as the distance between the centers of the cell C^k on frame k and cell C^{k+1} on frame $k+1$.

$$D_{displacement}(C^k, C^{k+1}) = ||\overrightarrow{C^k_{center} C^{k+1}_{center}}||$$

The C^k_{center} and C^{k+1}_{center} are defined as:

$$C^k_{center} = [(C^k_x + \frac{C^k_w}{2}), (C^k_y + \frac{C^k_h}{2})]$$
$$C^{k+1}_{center} = [(C^{k+1}_x + \frac{C^{k+1}_w}{2}), (C^{k+1}_y + \frac{C^{k+1}_h}{2})]$$

$D_{displacement}$ is the most basic parameter used to match the cell between two frames. It has no relationship with the topological features of the cell, only related to the movement features of the cell. It has been successfully used in feature point tracking in a high-density image [13]. But by experiments with some datasets, $D_{displacement}$ only works well in some sequence images in which cells move slowly and without too much cell overlap. It cannot deal with some complex sequence images so we define some topological features with better robustness such as $D_{circumference}$ and D_{area}.

The $D_{circumference}$ is defined as the circumference difference between the ellipse fitted cell C^k on frame k and the ellipse fitted cell C^{k+1} on frame $k+1$.

$$D_{circumference}(C^k, C^{k+1}) = |C^k_{circumference} - C^{k+1}_{circumference}|$$

The $C^k_{circumference}$ and $C^{k+1}_{circumference}$ are defined as:

$$\begin{aligned} C^k_{circumference} &= 2\pi * min(C^k_w, C^k_h) \\ &\quad + 4 * (max(C^k_w, C^k_h) - min(C^k_w, C^k_h)) \\ C^{k+1}_{circumference} &= 2\pi * min(C^{k+1}_w, C^{k+1}_h) \\ &\quad + 4 * (max(C^{k+1}_w, C^{k+1}_h) - min(C^{k+1}_w, C^{k+1}_h)) \end{aligned}$$

The D_{area} is defined as the area difference between the ellipse fitted cell C^k on frame k and the ellipse fitted cell C^{k+1} on frame $k+1$.

$$D_{area}(C^k, C^{k+1}) = |C^k_{area} - C^{k+1}_{area}|$$

The C^k_{area} and C^{k+1}_{area} are defined as:

$$C^k_{area} = \pi * C^k_w * C^k_h$$
$$C^{k+1}_{area} = \pi * C^{k+1}_w * C^{k+1}_h$$

We use these three parameters to compute the matching cost between two cells on frame k and frame $k+1$. Because the cell radius, image resolution, and cell movement speed of different cell image sequences are quite different, we set different weighting values to ensure the robustness of the method.

$$\begin{aligned} Cost(C^k, C^{k+1}) &= \alpha * D_{displacement}(C^k, C^{k+1}) \\ &\quad + \beta * D_{circumference}(C^k, C^{k+1}) \\ &\quad + \gamma * D_{area}(C^k, C^{k+1}) \end{aligned}$$

Since most of the cells change slightly between two consecutive frames, we chose the Hungarian algorithm [14] to minimize the total matching cost between two frames:

Total Matching Cost =

$$min\left(\sum_{C^k \in frame\, k, C^{k+1} \in frame\, k+1} Cost(C^k, C^{k+1})\right)$$

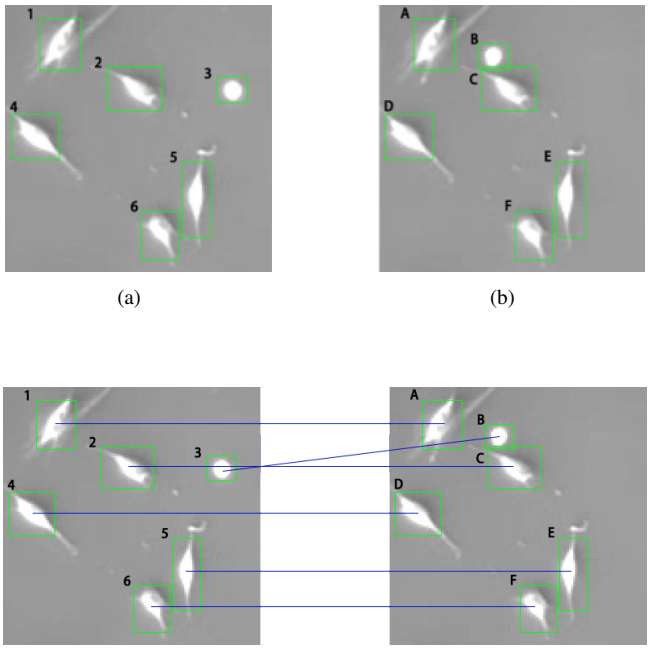
With constraint:

$$\sum_{C^k \in frame\, k} X_{C^k, C^{k+1}} = 1$$
$$\sum_{C^{k+1} \in frame\, k+1} X_{C^k, C^{k+1}} \leq 1$$

where $X_(C^k, C^{k+1})$ is a binary matrix and $X_(C^k, C^{k+1}) = 1$ if and only if we believe that C^{k+1} and C^k is the same cell. The constraint means that there must be one cell in frame $k+1$ that matches the cell in frame k no matter how much the matching cost is, but one cell in frame $k+1$ can only match no more than one cell in frame k, if there is no cell match in frame k we mark C^{k+1} as a new cell.

Because the number of cells in two consecutive frames is not the same all the time. Cell division and border enters will increase the number of cells on frame $k+1$, t border disappears

and segmentation error will cause some cells to disappear in the next frame. We set a threshold of matching cost, for C^k and C^{k+1} that $Cost(C^k, C^{k+1}) >$ threshold, they cannot be the same cell and we will drop this cell matching pair if the cost is larger than the threshold and mark C^{k+1} as a new cell.



(a) (b)

(c)

Fig. 7. Illustration of cell matching. (a) Segmented cell in frame t012.tif of PhC-C2DL-PSC Sequence 1. (b) Segmented cell in frame t013.tif of PhC-C2DL-PSC Sequence 1. (c) A synthetic image representation showing cell matching between two frame.

TABLE I
MATRIX OF MATCHING COST FOR CELLS

$Cost$	C_A^{k+1}	C_B^{k+1}	C_C^{k+1}	C_D^{k+1}	C_E^{k+1}	C_F^{k+1}
C_1^k	48.5	187.6	322.6	271.2	771.0	728.0
C_2^k	178.1	147.7	38.4	256.4	513.	550.0
C_3^k	244.6	204.3	247.6	527.0	92.1	512.0
C_4^k	267.1	322.9	24.1	28.4	439.	451.2
C_5^k	747.5	492.1	487.7	432.9	34.5	127.1
C_6^k	672.5	493.7	501.5	448.7	118.9	64.2

Figure 7 is an example of cells matching between two consecutive frames. Figure 7 (a) and (b) show the segmented cell in two frame, there are 6 cells and we labeled as $[1, 2, 3, 4, 5, 6]$ in frame 12 and $[A, B, C, D, E, F]$ in frame 13 respectively. The matching costs are listing in Table.1. As we can see, although the distance between cell 3 and cell C is smaller than the distance between cell 3 and cell B, by introducing topological features and minimizing total matching costs, our algorithm works well in this situation. The result of cell matching is illustrated in (d).

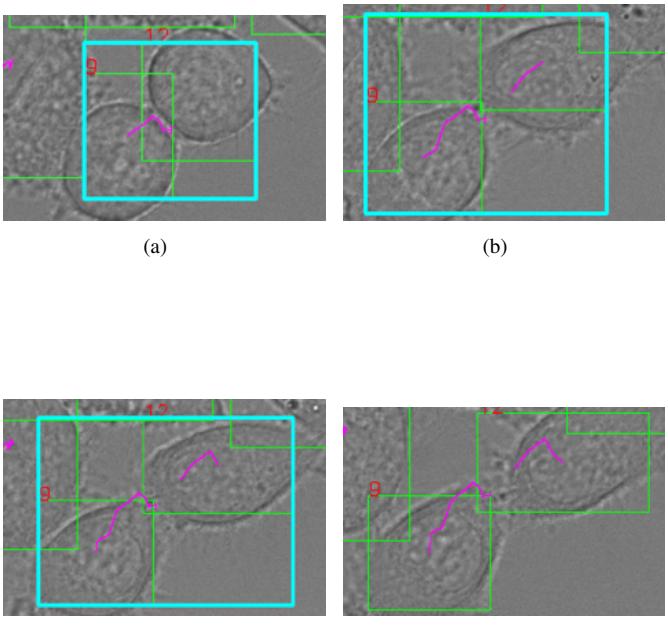
After matching all the cell between two consecutive frames, we can get the cells which do not exist in the previous frame. This is the basic basis for us to judge the possibility of division. In order to determine whether the new cell is produced by division, we need to find another cell that was produced by divided. As two cells that have just begun to divide, the distance between the centers of the two cells will not be too large. This can effectively narrow our search range. We set this threshold to three times the radius of the newly emerging cell. Thus, we can get:

$$D_{displacement}(C, C^{'}) < 3 * \sqrt{(\frac{C_w}{2})^2 + (\frac{C_h}{2})^2}$$

The C is the new cell and $C^{'}$ is the cell which is maybe produced by division. Then we need more constraints to select the only one cell which is produced by division. Therefore, we choose the areas of the cell, if cell C and cell $C^{'}$ are divided by the same cell, they should have a similar size. Thus, we choose the cell with the most similar area and the difference no more than 15% of C_{area}.

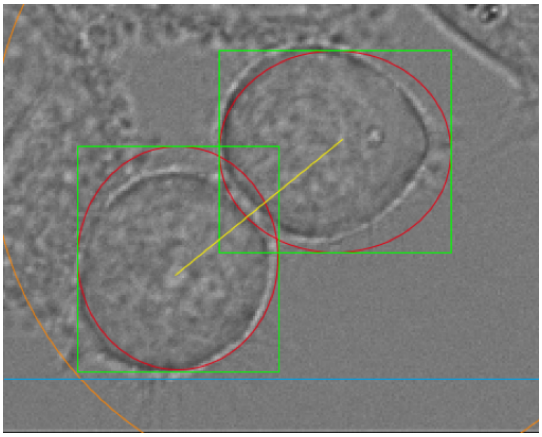
$$\min\{D_{area}(C, C^{'})\}; D_{area}(C, C^{'}) < 0.2 * C_{area}$$

Finally, in order to avoid being affected by the cells entering the edge of the image, we set a safety margin usually 5% of the edge of the image. For the cell which touches the safety margin is not considered for division. After marking as division cells, we will continue to compute their area changes and stop when their area changes less than 10% per frame.



(a) (b)

(c) (d)



(e)

Fig. 8. Illustration of cell division detect. (a) Begin of division in frame t011.tif of DIC-C2DH-HeLa Sequence 1. (b)-(c) Cell division in frame t012.tif and t013.tif of DIC-C2DH-HeLa Sequence 1.. (d) End of division in frame t014.tif of DIC-C2DH-HeLa Sequence 1 because the area changes are less than 10%. (e) Schematic diagram of division detection parameters.

Figure 8 is an example of cell division detect between two cells. Figure 8 (a) to (d) show a complete division process. Figure 8 (e) shows our parameter about division detecting: the bounding boxes are green, the $D_{displacement}(C, C^{'})$ is the yellow line, the threshold of distance is the orange circle, the areas of two cells are ren circle and the safety margin is the blue line.

C. Analyse Cell Motion

To further observe the movement of cells, a program that allows users to select a cell and get relevant information at any time is designed. This information includes the speed of the cell at the time point, the total distance that the cell has moved from the first time point to the current time point, the net distance between the first time point and the current time point where the cell is located and the ratio between the two distances.

1) Cell Selection: Function selectROI is a good method to select a cell at any time point, which implemented by looping through each image in the dataset and processing it accordingly.

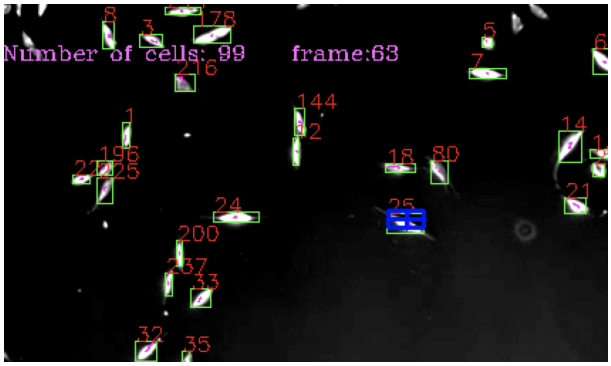


Fig. 9. Select a cell

Through this method, the relevant coordinate information of the box drawn by the user can get, including four parameters: x,y,w,h, and then based on the data from the previous part compute their centres:

$$x_{center} = x + \lfloor w/2 \rfloor$$

$$y_{center} = y + \lfloor h/2 \rfloor$$

where x_{center} and y_{center} are the coordinate x and y of the center point of the cell. After this operation, find the nearest bounding box to matching the corresponding cell by taking Euclidean Distance from the centre of the selected area to that of the bounding box of the cell:

$$Dis_2(x, y) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

2) Data Calculation: After getting the coordinate information of the selected cell, calculating the speed, the distances, and the relevant ratio of this cell by using Euclidean distance (in pixels) can be easily implemented. The first subtask is to compute the speed of the cell at any time point, which can be achieved by dividing the Euclidean distance between the center of the cell at the previous time point and the current one by the time difference. However, for the cell trajectory that appears for the first time, its speed cannot be calculated because there is no previous time point. If the coordinate of the cell at the current time point t is (x_t, y_t) and that of it at the previous time point $t-1$ is (x_{t-1}, y_{t-1}), the calculation is as follows:

$$v = \frac{\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}}{1}$$

where the denominator is 1 frame so that the unit of speed is pixels/frame.

In addition, the difference between the total distance and the net distance is that the first one is the cumulative result of the distance of the cell from the first time point of appearance to the current time point,while the net distance is just the Euclidean distance between the center of the cell at the first time point and the current time point.

$$Dis_{total} = \sum_{i=1}^{n} Dis_2(x_i, y_i)$$

$$Dis_{net} = \sqrt{(x_1 - x_i)^2 + (y_1 - y_i)^2}$$

where i means the current time point. Moreover, the ratio between the total distance and the net distance is based on the data form the previous subtasks:

$$ratio = \frac{Dis_{total}}{Dis_{net}}$$

Ultimately, output the above information in the terminal to facilitate observation and analysis of cell motion.

IV. Experimental Setup

A. Segementation

As we described in III-A1b, we need to make all cells erode. However, the parameter k chosen for the three datasets are different. Typically we choose smaller k for larger cells. Considering the rich-texture of cells in DIC-C2DH-HeLa adds difficulties for segmentation, we choose even smaller k.

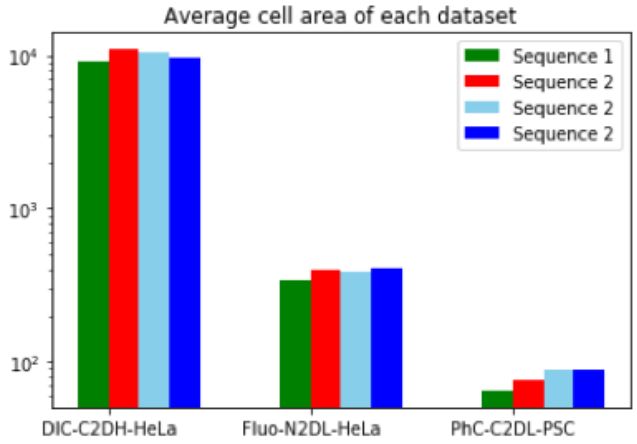
We also described data augmentation in III-A1c, so the range of scaling and rotating also needs to be set. The preprocessing parameters we set is shown here:

Dataset	k_{mask}	k_{marker}	scale range	rotation range
DIC-C2DH-HeLa	0.95	0.8	[0.6,1.4]	[-180°,180°]
Fluo-N2DL-HeLa	0.98	0.8	[0.6,1.4]	[-180°,180°]
PhC-C2DL-PSC	0.98	0.9	[0.6,1.4]	[-180°,180°]

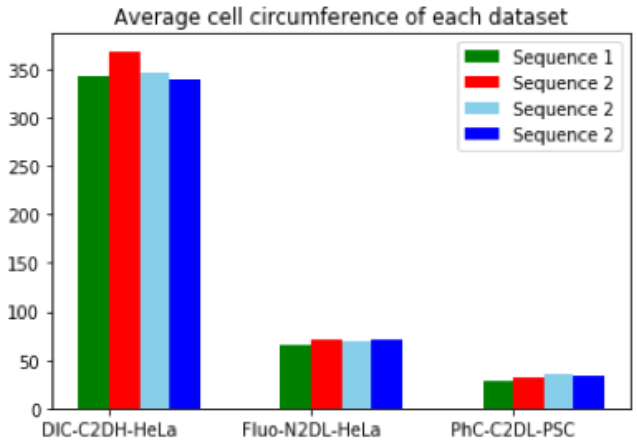
Since the output of U-Net model is probability of whether there is a cell at a pixel. The threshold we use is always 127.

In the postprocessing part, we also treat contours whose radius is smaller than one tenth of the radius of the largest contour in the frame as noise and delete them.

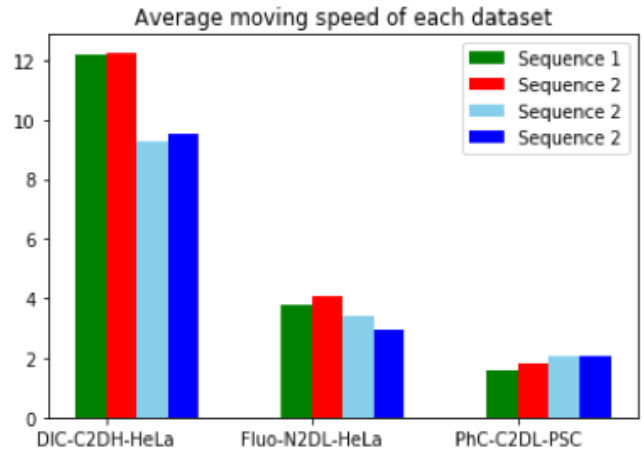
B. Tracking & Division Detecting



(a)



(b)



(c)

Fig. 10. Cell analysis of each sequence. (a) Average cell area of each dataset. (b) Average cell circumference of each dataset. (c) Average moving speed of each dataset.

Through the data of Figure 10, we have carried out multiple experiments on the weights of cell tracking. According to the experimental results, we have obtained the weights suitable for each sequence.

At the same time, according to data analysis, we obtained the max and min cell size and max movement speed of each sequence and set a series of feature matching and division detect thresholds from this.

V. Results and Discussion

A. Results of Detection

To evaluate the accuracy of cell detection, we randomly select N frames in each dataset and manually count the number of true positive(detected cells), false positive(recognize noise as cell) and false negative(undetected cells). Then, we calculate precision and recall for all datasets. For evaluating the model, we applied a **very strict counting regulation that we count one cell in as long as any part of the cell appears in the image.**

TABLE II
Detection Results

Dataset	N	Detected	Total	Precision	Recall
DIC-C2DH-HeLa	205	2394	2680	99.93%	89.33%
Fluo-N2DL-HeLa	131	12250	12629	100%	97.00%
PhC-C2DL-PSC	84	6621	7170	100%	92.34%

According to Table II, the precision is very high which means the algorithm is robust to noise. In the process of counting, we also notice that the precision does not decrease much when cell population massively increases. It indicates that the algorithm is also robust to cell density.

Moreover, **the prediction accuracy of two U-Nets are over 98%**. In this case, the recall score is unreasonably low. So we went over the process of segmentation and found that the background truth provided by CTC ignores cells near the edges of frames.

However, the real problem that still remains unsolved is over segmentation for quite large cells, which is shown here:

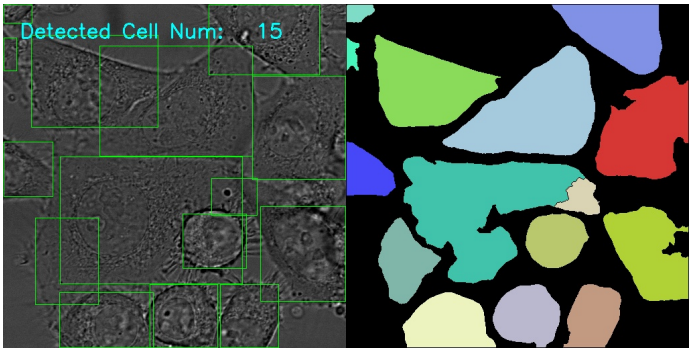
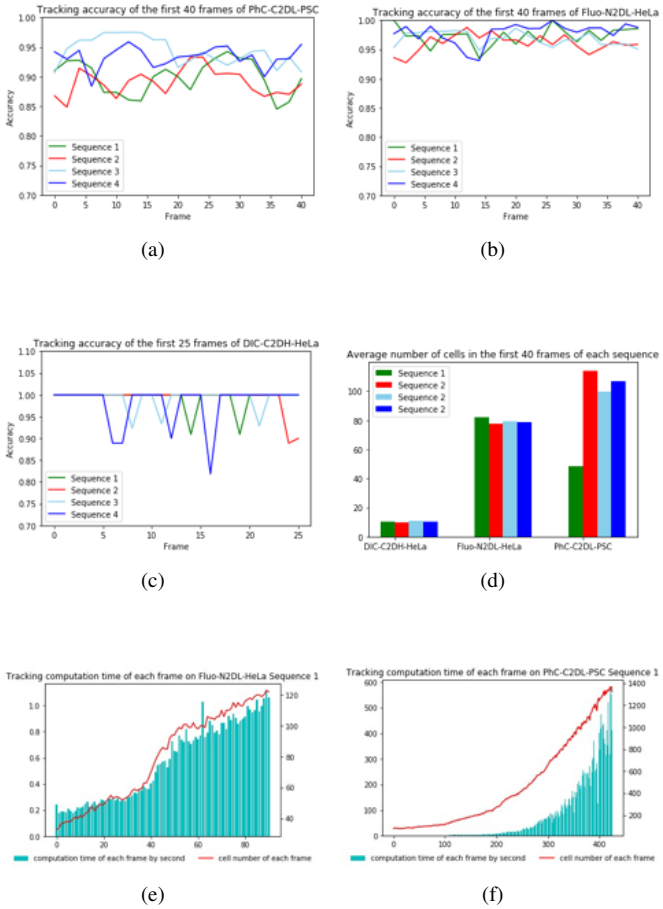


Fig. 11. Over segmentation for large cell

B. Results of Tracking



(a) (b) (c) (d) (e) (f)

Fig. 12. Result of evaluation of tracking. (a) Tracking accuracy of the first 40 frames of PhC-C2DL-PSC. (b) Tracking accuracy of the first 40 frames of Fluo-N2DL-HeLa. (c) Tracking accuracy of the first 25 frames of DIC-C2DH-HeLa. (d) Average number of cells in the first 40 frames of each sequence. (e) Tracking computation time and number of cell of each frame on Fluo-N2DL-HeLa. (f) Tracking computation time and number of cell of each frame on PhC-C2DL-PSC.

In the evaluation of the cell tracking effect, we use the ratio of the number of cells correctly tracked between two consecutive frames in the image sequence to the number of cells in the actual two frames. We define correct tracking as connecting the same cell in two frames with the same cell number.

According to Figure 12, our algorithm can keep about 90% of continuous frame tracking accuracy in each data set. Among them, the average accuracy rate in DIC-C2DH-HeLa is the highest. We believe this is because the number of cells in DIC-C2DH-HeLa is always small and the amplitude of cell movement is small. In the data sets of PhC-C2DL-PSC and Fluo-N2DL-HeLa with a large number of cells, the accuracy rate fluctuates greatly.

At the same time, we count the relationship between the computing time of the tracking algorithm and the number of cells. As shown in Fig. 12 (e) and (f), the calculation time increases as the number of cells increases. For images with a cell count of less than 100, the tracking can be completed within 1 second.

C. Results of division detecting

In the evaluation of the cell tracking effect, we use the ratio of the number of divisions correctly detecting in the image sequence to the number of divisions in the actual.

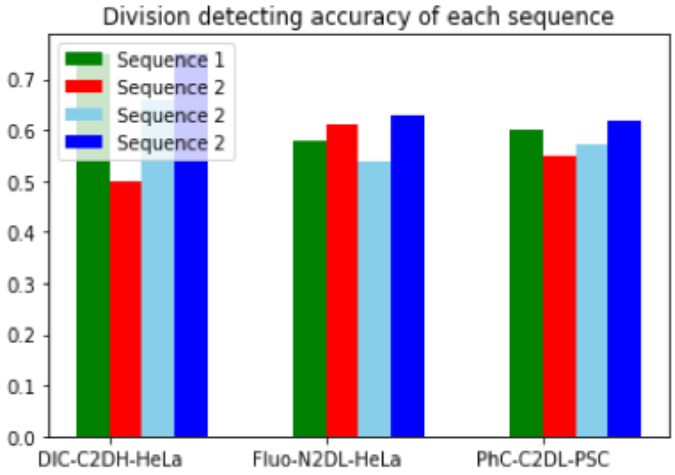


Fig. 13. Select a cell

According to the Fig. 13, our algorithm can only detect half of the actual divisions in each image sequence. This means that we still have a lot of room for improvement in the division detection algorithm.

VI. Conclusion

We have proposed a robust and automatic method for detecting and tracking cells, detecting cell divisions, and analyse cell motion. The proposed algorithm addresses the problems occurs in segmentation based detection due to noise, nonuniform illumination, uneven motion, low contrast and fluorescence. The method uses U-Net followed by a watershed segmentation algorithm for the aim of cell detection. This method focuses on the nucleus it self, so even for the different cell morphologies,

cells overlaps, mitosis, and cell-shape changes, this method is also effective. In the tracking and divisions detection method, we have applied a robust dissimilarity measure which combines cell movement features and cell topological features to improve the accuracy of matching. Even in uneven movement, cells clusters, or shape deformation, the accuracy of tracking with this method performs well. The experiments results aforementioned have showed that the proposed method obtains a high-quality performance and provides a high accuracy in detecting and tracking cells. In the cell motion analysis, ROI method has been introduced for selecting a cell at any time point and then analyse the motion of the selected cell. However, if the cells are on the boundary of the image and not integrated, or too small, the method may not be able to detect and track them. Besides, with the number of cells in the image increasing, the computing time of tracking may become longer. Therefore, we will further work on these two problems.

VII. Contribution of Group Members

Yangyang Liu: Design and imply the segmentation algorithm and write the corresponding part of the report. Present the online demo.

Wenbin Wang: Design and implementation of tracking and division detect algorithms and write the corresponding part of the report. Present the online demo.

Hongyu Lin: Define problems, analyzing given datasets and write the corresponding part of the report. Present the online demo.

Zhenzhang Wang: Responsible for a literature review and conclusion and write the corresponding part of the report. Present the online demo.

Nan Wang: Responsible for cell motion analysis and write the corresponding part of the report. Present the online demo.

References

[1] J. C. Caicedo, J. Roth, A. Goodman, T. Becker, K. W. Karhohs, M. Broisin, C. Molnar, C. McQuin, S. Singh, F. J. Theis *et al.*, "Evaluation of deep learning strategies for nucleus segmentation in fluorescence images," *Cytometry Part A*, vol. 95, no. 9, pp. 952–965, 2019.

[2] E. Meijering, O. Dzyubachyk, I. Smal, and W. A. van Cappellen, "Tracking in cell and developmental biology," in *Seminars in cell & developmental biology*, vol. 20, no. 8. Elsevier, 2009, pp. 894–902.

[3] E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert, and D. Van Valen, "Deep learning for cellular image analysis," *Nature methods*, pp. 1–14, 2019.

[4] J. W. Johnson, "Adapting mask-rcnn for automatic nucleus segmentation," *arXiv preprint arXiv:1805.00500*, 2018.

[5] H.-F. Tsai, J. Gajda, T. F. Sloan, A. Rares, and A. Q. Shen, "Usiigaci: Instance-aware cell tracking in stain-free phase contrast microscopy enabled by machine learning," *SoftwareX*, vol. 9, pp. 230–237, 2019.

[6] R. Hollandi, A. Szkalisity, T. Toth, E. Tasnadi, C. Molnar, B. Mathe, I. Grexa, J. Molnar, A. Balind, M. Gorbe *et al.*, "A deep learning framework for nucleus segmentation using image style transfer," *bioRxiv*, p. 580605, 2019.

[7] V. Ulman, M. Maška, K. E. Magnusson, O. Ronneberger, C. Haubold, N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic *et al.*, "An objective comparison of cell-tracking algorithms," *Nature methods*, vol. 14, no. 12, pp. 1141–1152, 2017.

[8] C.-M. Svensson, A. Medyukhina, I. Belyaev, N. Al-Zaben, and M. T. Figge, "Untangling cell tracks: Quantifying cell migration by time lapse image data analysis," *Cytometry Part A*, vol. 93, no. 3, pp. 357–370, 2018.

[9] Y. Li, F. Rose, F. Di Pietro, X. Morin, and A. Genovesio, "Detection and tracking of overlapping cell nuclei for large scale mitosis analyses," *BMC bioinformatics*, vol. 17, no. 1, pp. 1–15, 2016.

[10] F. Lux and P. Matula, "Cell segmentation by combining marker-controlled watershed and deep learning," 2020.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[12] A. Arora and T. Qazi, "Computer vision based tracking of biological cells-a review," 2014.

[13] C. Andrey and L. Andrey, "Tracking feature points: Dynamic programming algorithm," in *2009 IEEE Congress on Evolutionary Computation*, 2009, pp. 1032–1037.

[14] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109