

Applied Computer Vision Methods in Segmenting and Tracking of Biological Cells in Time-lapse Microscopy Images

COMP9517 Computer Vision 2021T3 Group project

Mingxuan CHEN

group of YouISee

COMP9517 Computer Vision

Student ID: z5330738

z5330738@ad.unsw.edu.au

Jia JI

group of YouISee

COMP9517 Computer Vision

Student ID: z5272037

z5272037@ad.unsw.edu.au

Zhiyi ZHANG

group of YouISee

COMP9517 Computer Vision

Student ID: z5306801

z5306801@ad.unsw.edu.au

Guanzhu HOU

group of YouISee

COMP9517 Computer Vision

Student ID: z5325417

z5325417@ad.unsw.edu.au

Zheng FU

group of YouISee

COMP9517 Computer Vision

Student ID: z5285691

z5285691@ad.unsw.edu.au

Abstract—This paper implement an algorithm to segment and track cells from a sequence of continuous images, then analyze cell motion to obtain further information for research. For segmentation, we use global threshold for binarization, watershed and connected domain. And for cell tracking, we introduced several cell features and combine them into different rules to determine the relationship of cells in continuous images. Detect the occurrence of mitosis. Then track and analyze the motion of cells. Through our project, we can clearly identify cells in unclear images and performance more than half of the accuracy rate of cell motion tracking. Effectively collect information of given sequence of cell image.

Index Terms—cell tracking, cell segmentation, cell motion analysis, connected domain, OpenCV, feature matching.

I. INTRODUCTION

Nowadays, biology has been a field containing many interdisciplinary knowledge, so in bio informatics, computer vision technology is one of the most effective cutting-edge techniques in this area. Therefore, in this project of computer vision course, our task is to analyze several sequences of cell images observed by microscope in order to segment and track each cell. The completion of this project not only gave us a deeper understanding of computer vision and image processing, but also let us understand the content of cell analysis in biology.

As we all know, there are a lot of information contained in microscope image of cells, but some of them are not very obvious and a bit hard to analyze. If there is an algorithm which can produce such information effectively, researchers could describe cell activities easily and in more detail, such as movement, mitosis, phagocytosis, apoptosis, etc. Analyzing cells based on manpower is in efficient and lack of stability for it is dependent a lot on personal experience, which is difficult to be widely used. Therefore, more and more cell detection

and tracking technique has been developed in computer vision area.

In cell tracking, threshold method is the wildly used and easiest to implement. But, precisely because this method is very simple, it shows a low level of compatibility for images with different quality, which need to be manually adjusted to a suitable threshold value. As well, preprocessing requirements for images are relatively high for threshold method, because when the contrast between the cells and the background is no very high, it cannot segment them successfully. Relationships among cells are also be ignore in this method. Cells which are very close could be recognise as a large cell mass due to threshold method focus on separating cells from background. In addition, watershed method could identify group cells based on their grayscale value. But it may appear too finely divided in watershed. Instead, template matching is a good method its need a high requirement of implementation. This method need to establish a model with cell morphology and base on it we can get a good result of segmentation.

II. LITERATURE REVIEW

A. Cell Segmentation

Cell detection and classification is the basic task in the analysis process. Cell detection mainly includes threshold method, edge detection, pattern matching and so on. First, for high contrast images, such as sample images with fluorescent labeled cell membranes, cells can be detected very quickly by setting pixels above the threshold as the foreground and pixels below the threshold as the background.

The second common method is edge detection. When there are extreme intensity changes in the image, the changing area is the edge area of the cell. (Gonzalez, 1977) described in the

book *Digital Image Processing* that energy is to use Fourier transform to analyze the severity of changes in the image and convert the image from spatial domain to frequency domain. Then, the edge detection of cells can be realized by using the edge described by gradient change. The disadvantage of this method is that if the edge intensity changes little, the detection effect is not good, and there is optical blur in the fluorescent image during microscope shooting, it can not be avoided. At the same time, the same disadvantage as the threshold method is that when the cells overlap or are adjacent, the previous segmentation of cells cannot be processed.

The third common method is artificial intelligence, such as U-net, which is a very successful neural network model. And U-net has achieved the best results. Its method is to use the neural network of encoder and decoder to segment the cell image semantically. U-net is a convolutional neural network with 26 hidden layers. It divides the low-resolution large-scale structure into high-resolution small-scale structure, and trains the annotated model, which can facilitate migration learning and reduce the training time and iteration times from scratch.

B. Cell Tracking

The algorithm of tracking can be roughly divided into several directions. First, based on mean shift algorithm. The projection of similar regions in the histogram of cell image can obtain the regions with high density in the image, and the vector sum of sample feature points can be calculated continuously until convergence. This iterative process can obtain the mean shift vector between two frames. When the value is less than the threshold, it represents the moving path. But this method cannot deal with cell mitosis. Secondly, the basic idea of the adjacent inter frame difference method is to use two or several adjacent frames to make a difference, and the obtained difference image is the detection result. Third, optical flow tracking seeks the relatively changed optical flow field in the constant optical flow field, and the changed part is the part where the moving target is located. There is also the target tracking realized by active contour and Kalman filter. In addition, particle filter is a process of approximately representing the probability by looking for a group of random samples propagating in the state space, so as to obtain the minimum variance estimation of the system state. Of course, these methods are based on strong cell model assumptions.

III. METHODS

Justify and explain the selection of the methods we implemented, it will divided into three parts.

First part is about cell segmentation, it will explain how we preprocess the data image and show the contours for the cells; Second part is about cell tracking, it will explain how we track the cell in different image and show their trajectories and find if the cell is in the process of dividing; third part is about analyze cell motion which will explain the data we obtained.

A. Cell Segmentation

To begin with, we should load the picture and turn it into a grayscale image. Using the sobel operator to calculate the

gradient in x and Y direction, and then subtract the gradient in Y direction from x direction. Through this subtraction, we leave the image region with high horizontal gradient and low vertical gradient.

To remove the noise on the image, the image is first smoothed using a low-pass filter (9 x 9 kernel), which will help smooth the high-frequency noise in the image. The goal of the low-pass filter is to reduce the rate of change of the image. For example, replace each pixel with the mean of the pixels around it. This smoothes out and replaces areas where the intensity varies significantly.

Then, binarize the blurred image. Any pixel in the gradient image not greater than 90 is set to 0 (black). Otherwise, the pixel is set to 255 (white).

After that, we find there are some small white spots on the image, which will interfere with the detection of the cell contour later. They should be removed. Based on morphological erode and dilate operation we performed morphological opening and closing operation to the image. Then use a GaussianBlur with a 3 x 3 kernel, in order to remove the incomplete cells on the boundary of the image, we used the mohotas Library. After use Remove bordering function, the incomplete cells on the boundary were removed successfully. Next, set the threshold to 129, finally perform the watershed to finish the preprocessing part.

After preprocessing part, next step is to find the outline of each cell. There are severral parameters in *cv2.FindContours* function, the first parameter is the image to be retrieved, which must be binary, i.e. black and white (not grayscale). Therefore, the image to be read should be transformed into gray scale first, and then into a binary graph. The binary graph has been obtained in the preprocessing part. The second parameter represents the retrieval mode of the contour, there are four kinds we choose the *RETR-TREE* which creates an outline of the hierarchical tree structure. The third parameter is the approximate method of contour:

$$\text{Max}(\text{abs}(x_1 - x_2), \text{abs}(y_2 - y_1)) = 1 \quad (1)$$

CHAIN-APPROX-SIMPLE Compresses elements in horizontal, vertical, and diagonal directions, keeping only the end coordinates of that direction. For example, a rectangular outline needs only 4 points to store the outline information. The *cv2.FindContours* function returns two values, one for the contour itself and one for the properties of each contour. The *cv2.FindContours* function returns the first value as a list, where each element is an outline in the image, represented by ndarray in Numpy. Each ndarray saves the coordinates of each point on the outline. After shifting the result of watershed to a proper input of *cv.FindContours*, we find the connected domain and operate on the connected domain. If the area of the connected domain is bigger than the min cell area set by us, we will collect this domain, otherwise we will ignore it. Draw contours on the image in OpenCV using *cv2.DrawContours*: The first parameter specifies which image to draw the outline on. The second argument is the outline itself, which in Python is a list. The third argument specifies which outline in the list

to draw, or if -1, all of it. The fourth parameter is the color of the outline. The fifth parameter is the thickness of the outline.

We use the *MinAreaRect* function which output the rectangle containing the minimum area of the point set is mainly obtained. This rectangle can have deflection Angle and could not be parallel to the boundary of the image. Then compute the rotated bounding box of the largest contour and draw a bounding outline arounded the detected cell and display the image.

B. Cell Tracking

After the accomplishment of the cell segment part, a temporal dimension weakly linked model is generated, where cell segments in each image will be calculated and updated independently. Now the issue moves to the association of processed segments from image to image and strengthens connections. [1]

We used to download the cell tracking tool MTrackJ on ImageJ. However, the python interface of ImageJ does not contain plug-ins downloaded by users themselves. Then we pick the approach that associates each cell in the current image to the spatially nearest cell in the old frame. And to reduce the case of mismatch, we combine the nearest-neighbor linking approach with the features of the cells. [2] After experiments, the best one is finally selected from the seven combinations.

Here are the details of the matching and tracking process:

Based on the contours obtained in the segment part, we first perform the calculation of cell features and get the cell information such as cents, areas, perimeters, as well as intensities. The intensities are the total pixel value of contour points in the original image of the gray scale divided by the number of contour points. With the support of Euclidean distance, we find the two closest cells **min** and **sec** in image t to a cell **old** in image $t - 1$.

After that, we define two relationships between the cells in two consecutive images. One is parent-child, where parent cell produces children cells after mitosis. The other is predecessor-successor, where the cell in the old image is not in mitosis and appears in the next image.

If the ratio in Eq. (2) and Eq. (3) are both within split range and even the distance between **sec** and **old** is closer than the distance threshold, we assume that mitosis is now underway and the relationship between these three cells is parent-child. [3]

$$\text{ratio} = \frac{\text{the area of } \mathbf{min}}{\text{the area of current cell } \mathbf{old}} \quad (2)$$

$$\text{ratio} = \frac{\text{the area of } \mathbf{sec}}{\text{the area of current cell } \mathbf{old}} \quad (3)$$

If **old** is not in mitosis, We initially designed five methods to judge whether it belongs to predecessor-successor relationship. As what is shown in Table I, we compare **min** and **sec** with **old** and take different cell features every time. We also add the restriction of Euclidean distance.

For type 1, we only use **min**. And if the distance between **min** and **old** is closer than the distance threshold, and the area ratio in Eq. (2) is within similarity range, we assume **min** in

TABLE I
JUDGEMENT OF PREDECESSOR-SUCCESSOR RELATIONSHIP

Type Number	Cell Features and Involved Cells		
	Cell Features	Cell min	Cell sec
1	area ratio ^a	✓	
2	intensity ratio	✓	
3	area ratio	✓	✓
4	0.5×(intensity ratio+area ratio)	✓	✓
5	0.5×(intensity ratio+area ratio)	✓	

^a t feature divided by $t - 1$ feature

image t is the successor of **old** in image $t - 1$. For type 2, the similarity is certificated by the intensity of **min** divided by **old**. For type 2, if the distance between **sec** and **old** is closer than the distance threshold, and both area ratios of **min** and **sec** are within similarity range, we select the cell with the smallest absolute value of the distance from 1 as the successor of **old**. The only difference between type 4 and 3 is type 4 focuses on weighted similarity ratios. For type 5, it applies the same approach as type 4 to check similarities but only uses cell **min**.

We also test with more cells (3 and 5) in image $t - 1$ with the same way of getting similarities in type 4 and 5. For the multiple closest cells (3 or 5), of which distance from **old** is less than the distance threshold, we pick the one with the smallest absolute value of distance from 1 as the successor of **old**. Hence, we have 7 ways of judgement on predecessor-successor relationship on total.

Cells that are neither in parent-child nor predecessor-successor will be deleted as new cells and registered in image t . Once the appropriate associations within the successive image is implemented, we could continuously analyze and discuss the cell motions. The centroid of each cell will be saved and the chronological connection of centroids will be regarded as the trajectory of a cell.

C. Analyze Cell Motion

As the method above, the number counting cells is considered as the contour discovered through built-in *cv2.findcontour*, which is implemented by Suzuki's Contour tracing algorithm [10], including outer boundary or the hole boundary. In this project, the cell finding only extracts the outermost border as discovery of one complete cell. In other words, the number of contours is equivalent to the number of cells in a image. Particularly, incomplete cells in the view should not counted, which can be considered as the cell touching edge pixel should be removed, which is resolved in pre-processing section.

Furthermore, based on the contour number calculation, the cell size on average can be derived by the formula below.

$$\bar{\text{size}} = \frac{\text{total cell area}}{\text{cell number}} \quad (4)$$

In the formula, the total cell area can be treated as the total contour area, by calling *cv2.contourArea*, which is computed using the Green formula [11], but it should be noticeable that

the function sometimes may calculate with wrong answer for contours with self-intersections.

In terms of displacement, in the above approach, these cell attributes are stored in a class file and matching algorithm as above mentioned is performed through iteration over all cells the other frame as prerequisite for locating identical cell over frames. According to cell association by ID assigned frame by frame, it is applied for searching two nearest cells of its previous after calculating centre pixel by moment, which are the average of the intensities of an image's pixels in openCV context, to judge mitosis status. After that, cell trajectory algorithm is applied by drawing movement of cell center.

IV. EXPERIMENTAL RESULT

Explain the experimental setup we used to evaluate the performance of the developed methods and the results we obtained.

This project based on python 3.8.8 and opencv. Significant Python packages are: mahotas, Numpy, OpenCV, SciPy, scikit-image.

The input data path and preprocess method can be changed in param.py.

Run the main.py and the system will start to read and process the image data, final result and analyze result will store in 'image data file name + gen' and 'image data file name + analysis'. We take Sequences/03 as example to show the result.

A. Evaluation of Segmentation

Original data image is Fig. 1. The result process by our cell segmentation method show in Fig. 2, Fig. 3, Fig. 4, Fig. 5, and final result is show in Fig. 6.



Fig. 1. Original data image

B. Evaluation of Tracking

After experiments, using 1 cell could perform well on matching (in Fig. 7), while using cells more than 1 will even make some mistakes. They will draw trajectories for several cells in the very beginning image. The circumstance is even

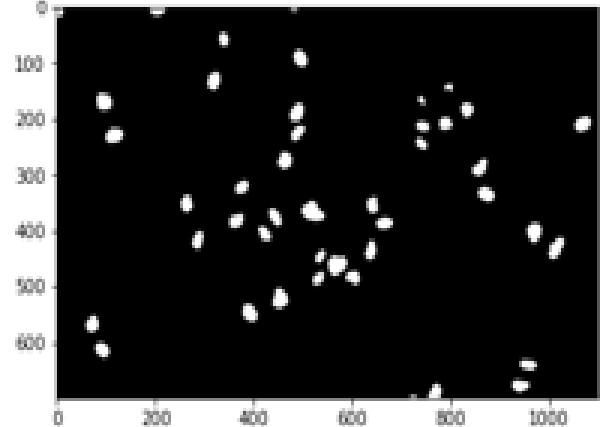


Fig. 2. Preprocess 1

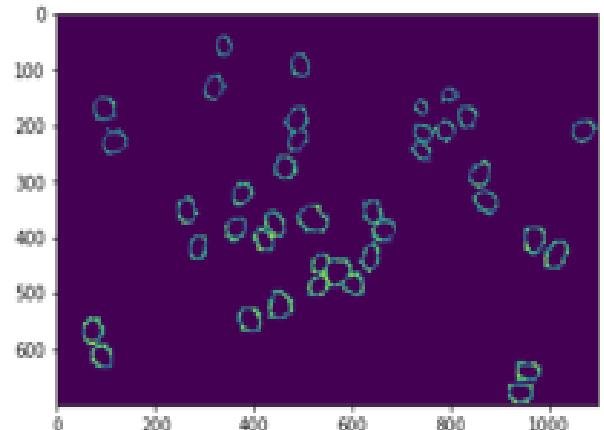


Fig. 3. Preprocess 2

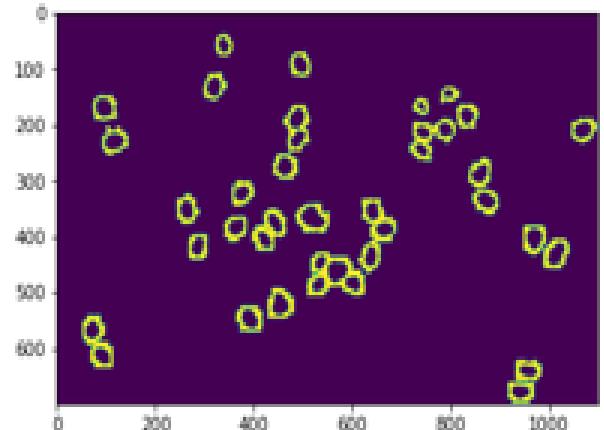


Fig. 4. Preprocess 3

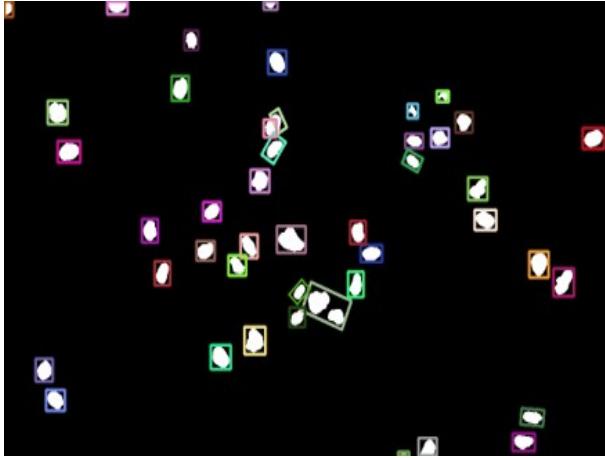


Fig. 5. Preprocess 4

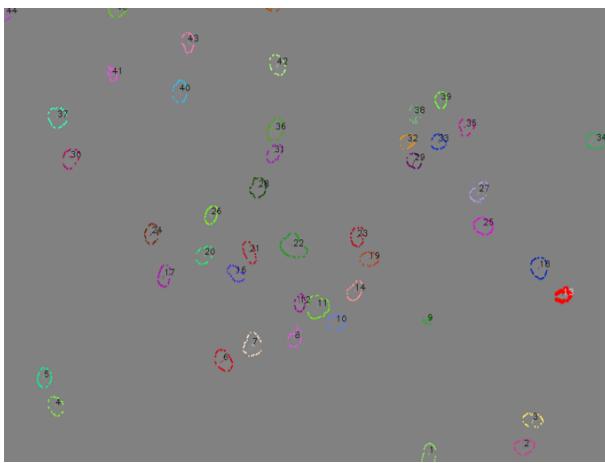


Fig. 6. Final result

worse when using more cells, which is shown in Fig. 8 and Fig. 9

And type 5 uses the weighted use of two features, which can make the model more robust. Thus the recommended type in params.py is type 5.

Perhaps this error is due to the fact that cells are registered many times in the process of comparison, and each registration will record a centroid position. If a cell records multiple centroid positions in image $t - 1$, there will be many trajectories in image t . In the future, it can be optimized by limiting that a cell can only be registered once and modifying the if statement to judge the relationship.

C. Evaluation Cell Motion Analysis

Cell count, average size of all cells, average displacement of all cells and the number of cell dividing will be described and counted in green font above the resulting picture. After the user queries a specific cell with ID, the information like perimeter, intensity, similarity, net distance traveled, total distance traveled and confinement ratio of the current cell will be described in red font above the green font (shown in Fig. 10)

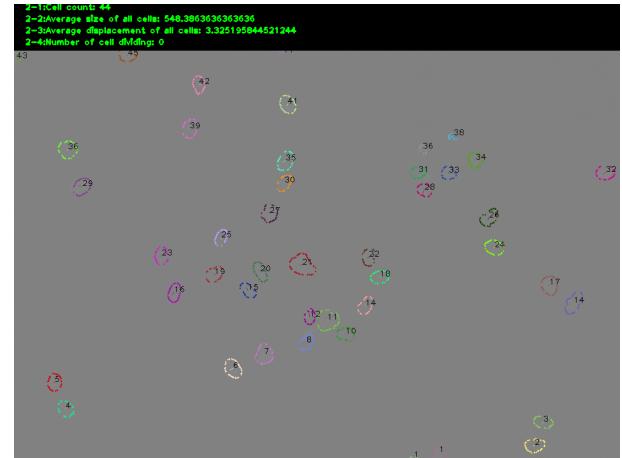


Fig. 7. cell track result with type 1

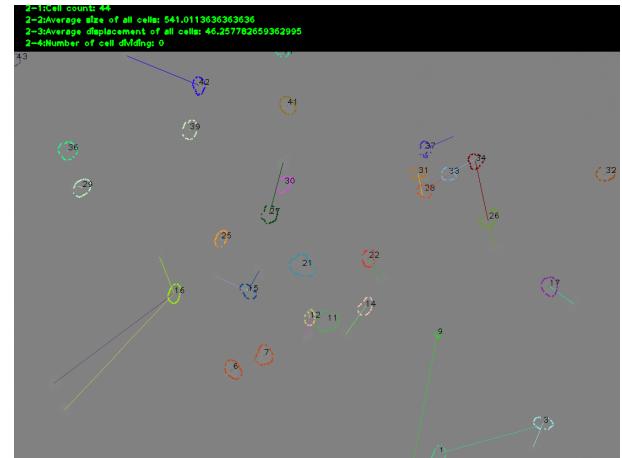


Fig. 8. cell track result with 3 points



Fig. 9. cell track result with 5 points

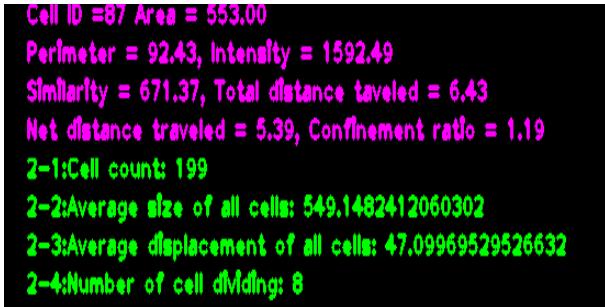


Fig. 10. Cell motion analysis

V. DISCUSSION

Our experiment displayed a good performance for cell tracking. Then in this section we are going to discuss the advantage and some future work.

A. Cell Segmentation

In segmentation procedure we introduce four types of image pre-processing and fully compare the result in order to find the type with best performance. Almost all noise in the image was successfully removed which generate a clean image for cell tracking procedure.

However, in boundary cell removing procedure, Gaussian filter will enlarge cells which cause closed ones to touch with each other. This filter could be improved with a better kernel or just find an alternative method to remove those cells.

Currently the segmentation in our experiment is fundamentally using basically computer vision method. During the research we also find some artificial intelligence method which shows good performances such as U-net introduced by Prof. Ronneberger [4]. It is a conversational network which is design and proven effective for biomedical image segmentation. That will be a good improvement to our project.

B. Cell Tracking and Motion Analysis

During tracking cells, we identify cells with contour and directly give them a certain ID and a unique color. These feature will follow cells in different images so that we can matching them efficiently and also can figure out a clear information for users.

However, we already used several features of cell to distinguish them between frames. For this procedure, there are two things still can be improved. Firstly, more features could be involved. Concavity of cell shape could help the detection of mitosis in our expectation. During mitosis, the cell could transform from convex to concave and then become two convex child cell. Besides, morphology, surface and curvature will also improve the tracking [5].

In addition, more frames could be considered in cell tracking instead of just two. For the procedure of mitosis is always a multi-frame motion, involve multiple images could increase the accuracy of mitosis judgement and tracking.

VI. CONCLUSION

This report introduce a pipeline to track and trace a group of cells in a continuous series of frames. For cell recognition and segmentation, we implement global threshold, Gaussian blur and watershed to obtain a binary image which display each cell clearly. Cells on the boundary is removed to deduct the affect to further steps. Then finding connected domain and draw the contour for each cell. To trace cell motion, we introduced area, perimeters and intensity as cell features. By using different combination of them, we design several type of judgement standard to determine the relationship of cells in continuous images. As a result, we can obtain the information and track the motion of cells.

VII. CONTRIBUTION OF GROUP MEMBERS

Chen Mingxuan. Implement cell segmentation part and use library Mahotas to implement boundary cell removing. Take charge of cell segmentation part in presentation and report.

Fu Zheng. Study several paper and implement experiment program. Implement part of prepossessing in program. Take charge of motion analysis part in presentation and experiment result in report.

Hou Guanzhu. Build the main pipeline of the program, implement cell tracking part. Present cell tracking part in presentation and take charge of that content in report.

Ji Jia. Study several paper and implement experiment program. Present introduction and overview in presentation. Take charge of introduction and literature review in the report.

Zhang Zhiyi. Study several paper and implement experiment program. Try the method of U-net. Take charge of future work part in presentation and discussion and conclusion part in report.

REFERENCES

- [1] Erik Meijering, Oleh Dzyubachyk, and Ihor Smal, Chapter nine - Methods for Cell and Particle Tracking, 2012
- [2] Erik Meijering and Oleh and Dzyubachyk and Ihor and Smal and Wiggett, Tracking in cell and developmental biology - ScienceDirect, 2009
- [3] Raymond Lu, Xiaocong Chen, COMP9517 Computer Vision 2020 Term 2 Project Cell Segmentation and Tracking, https://github.com/raymondlu1995/COMP9517_Cell_Segmentation_and_Tracking/blob/master/paper.pdf, 2020
- [4] Falk. T, Mai, D., Bensch, R. et al. U-Net: deep learning for cell counting, detection, and morphometry. Nat Methods 16, 67–70 (2019). <https://doi.org/10.1038/s41592-018-0261-2>
- [5] Masuzzo P, Van Troys M, Ampe C, Martens L. Taking aim at moving targets in computational cell migration Trends Cell Biol., 26 (2) (2016), pp. 88-110, 10.1016/j.tcb.2015.09.003
- [6] Vigny My, Identify objects and cropping in the picture by openCV, https://blog.csdn.net/qq_38660394/article/details/80609323
- [7] Mahotas, Labeled Image Functions, <https://mahotas.readthedocs.io/en/latest/labeled.html>
- [8] Luo D , Daly C J , Smart P , et al. Iterative Multilevel Thresholding and splitting for segmentation of 3D laser scanning confocal microscopy fluorescence images. 1999.
- [9] Debeir O , Ham P V , Kiss R , et al. Debeir O, Van Ham P, Kiss R, Decaestecker CTracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. IEEE Trans Med Imaging 24:697-711[J]. IEEE Transactions on Medical Imaging, 2005, 24(6):697-711.
- [10] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," Computer Vision, Graphics, and Image Processing, vol. 29, no. 3, p. 396, 1985.

- [11] Contour features, OpenCV. [Online]. Available: <https://docs.opencv.org/4.x/dd/d49/tutorial-py-contour-features.html>. [Accessed: 19-Nov-2021].