COMP9517 Project Report

1. Introduction

The task of this project is to trace biological cells in time-lapse microscopy images. As explained in the specification, this is an important step in many biological studies. As well-segmented images can better expose the morphological information of cells which have great biological significance. Such cell tracking is an important step in research. The traditional approach to this problem is manual annotations which is time-consuming and requires the expertise and professional knowledge. With the development in both computer vision and artificial intelligence, the fully-automated methods for cell tracking is possible. Although there have been many attempts to create better methods, the challenge still remains. Images with overlapping cells or images that have very complex structures may not reach the quality of manual annotations.

To achieve this task, steps including preprocessing, segmentation, tracking should be taken into consideration.

Computer visions such as watershed and template matching tend to produce relatively satisfying results. However, in images with increasing complexity, it becomes complicated for those methods to perform well. Deep-learning based methods such as U-net have recently been proved to produce good results. The U-net model requires training before being put into practical use. The training process includes input images and target images. Target images also require annotations to be able to train the model. The U-net models can perform well in complex images.

The key to having satisfying results is the combination of both deep-learning based methods with traditional methods.

The first dataset is "DIC-C2DH-HeLa" which contains sequences. Sequence 1 and 2 has 84 images. Sequence 3 and 4 has 115 images. All images are in grayscale images with the size of 512 * 512 pixels.

The second dataset is "Fluo-N2DL-HeLa" which has 4 sequences. All 4 sequences contain 92 images each. All images are grayscale images with the size of 1100 * 700 pixels.

The third dataset is "PhC-C2DL-PSC" which has 4 sequences. Sequence 1 and 2 contains 426 images. Sequence 3 and 4 contains 300 images. All images are grayscale images with the size of 720 * 576 pixels.

2. Literature Review

Since cell tracking is a popular topic, there have been a large number of published papers explaining different methods.

As explained in [1], cell tracking methods generally consist of two main image processing steps, namely, cell segmentation and cell association. The authors recommended watershed method, deformable models in the part of cell segmentation. These two methods are considered useful for this project while normal segmentation based on a preset threshold fails to segment cells that have intersections with each other. In the part of cell association, the authors mentioned associating each cell in any frame to the spatially nearest cell in the next frame within a predefined range. However, the concept of "nearest" must be discussed further. The key points of locating the "nearest" cell includes finding the closet matching cells in both distance and shape. This paper provided a well-established starting point for this project and forms the basis as well.

The article [2] discussed methods for Cell and particle tracking in greater depth. In particular, the authors provided ideas for cell linking, which the previous article authors refers to as cell association. The authors recommended template matching, mean-shift processing, or deformable model fitting. Once these methods are applied to one frame, positions or contours found can be used to initialize the segmentation process in the next frame, and so on, which implicitly solves the linking problem.

In addition, the authors also listed a few software tools that may be useful. However, the general assumption made by most cell tracking tools is that the cells can be modelled as bright regions against a darker background. If this is not the case, or if the images are too noisy, it is necessary to apply suitable filters to match this assumption. In the case of the dataset given, preprocessing seems necessary, especially for “DIC-C2DH-HeLa”. This paper therefore provided possible ways to solve the problem, but it would require preparation work to make full use of the methods recommended.

The article [3] focuses on detection and tracking of overlapping cell nuclei. Cells overlapping each other is a problem that normal watershed methods cannot solve. The authors discussed a 4-step approach to achieve the desired result. Among those four steps, GMM (Gaussian Mixture Model) is the critical component. The approach starts with localizing the fibronectin patterns and cropping the whole video at those locations to obtain individual cluster sequences, and applying 2- and 3-components GMM onto each frame of each video sequence. Then identify the first frame containing three cells using the fitting error difference and other features computed from the GMM parameters. The final step consists in the computation of the angle of division in the identified frame. The code is freely available to use and the results are satisfying. The authors claim that there is no existing software that could achieve the same result.

The article [4] introduced a generic deep-learning solution that can be applied to this project called ‘U-net’. U-net is a convolutional network containing 26 hidden layers that can be logically grouped in two parts. The first part is to yield a multilevel, multiresolution feature representation from input images. The second part classifies all pixels at original image resolution in parallel based on the features. The images shown in the article appear to have good results even if the images have complex and overlapping objects. The authors mentioned that the transfer learning on U-net requires considerably fewer annotated data than training from scratch. This is a very promising method that can be applied into this project.

The article [5] provided the possible best solution to the cell tracking problem by combining Marker-Controlled Watershed and deep learning. The images illustrated in the article showed satisfying results when segmenting touching and overlapping cells. The deep learning approach selected was CNN. The authors used one network to predict cell marker pixels, and the second network to predict the image foreground. The first network predicts for each pixel a probability of being a marker. The predictions are later fed into watershed marker function. The authors transformed the foreground prediction calculated by the second network to the watershed segmentation function and a mask of cell regions. Watershed segmentation is then used based on the marker function and Segmentation function. As the authors used datasets very similar to the ones given, this paper provides feasible ways to better segment and track touching cells. However, the choice of deep learning methods may differ in this project.

3. Methods

Figure 1 showed a flowchart of the experimental methods. Due to the difference between the three datasets given, there are two methods, namely, deep learning and traditional morphological operation have been chosen to detect and segment the cells.

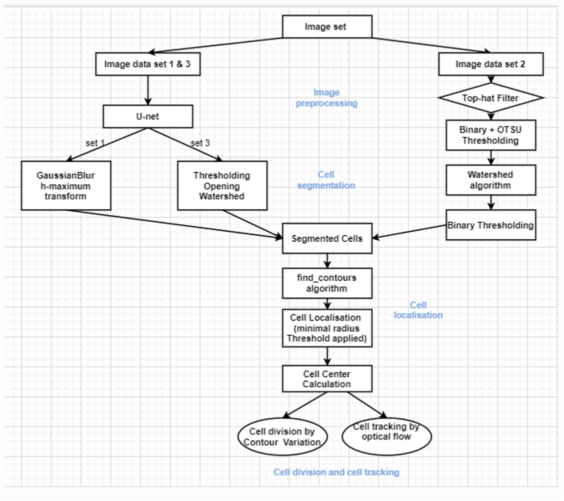


Image set
Image data set 1 & 3
Image data set 2
Image preprocessing
Top-hat Filter
U-net
set 1
set 3
Binary + OTSU Thresholding
GaussianBlur h-maximum transform
Thresholding Opening Watershed
Cell segmentation
Watershed algorithm
Binary Thresholding
Segmented Cells
find_contours algorithm
Cell Localisation (minimal radius Threshold applied)
Cell localisation
Cell Center Calculation
Cell division by Contour Variation
Cell tracking by optical flow
Cell division and cell tracking

figure 1.

As shown above, the U-net model of deep learning is used for DIC-C2DH-HeLa and PhC-C2DL-PSC. A Morphological operation combining with watershed is used for Fluo-N2DL-HeLa.

3.1 Cell Segmentation

3.1.1 Image preprocessing

For “Fluo-N2DL-HeLa”, data preprocessing is applied first, which aims to eliminate the dark shadows on the images caused by defects on the camera lens or the lighting issues. In this project, the method of Top-hat filter have been used to reduce these visual artifacts.The aforementioned operations can be formulated as

$$\hat{A} = top\text{-}hat(A', B) = A' - (A' \circ B)$$
$$= A' - max_B(min_B(A'))$$

In addition, a watershed method combined with a threshold is applied. The threshold used is a combination of both Otsu and Binary thresholding. The kernel of the filter is 3*3 which is determined by the structuring elements of an ellipse. The minimum filter and maximum filter are generated using eroding and dilating the original image with the kernel. Then the Euclidean distance from every binary pixel to the nearest zero pixel is computed to find peaks. Watershed model is applied to perform connected component analysis at last. After performing arithmetic operation between the filters and the original image, the results are normalized using OpenCV functions before segmentation.

3.1.2 U-net processing

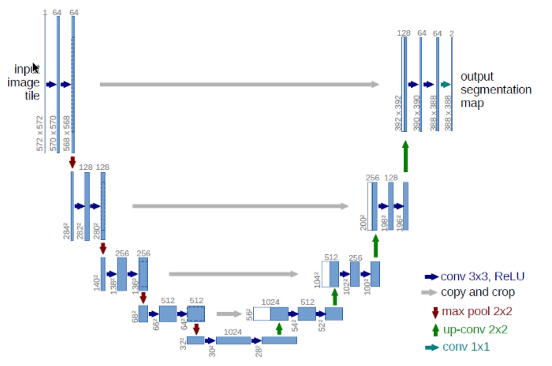


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

figure 2. image of U-net structure

Method used for DIC-C2DH-HeLa and PhC-C2DL-PSC is a pre-trained U-net model from deepimgaej on github [7]. Figure 2 has shown a completed U-net structure, the model was trained based on the data provided by Dr. T. Becker. Fraunhofer from Germany. The U-net model serves to predict the foreground pixels. To further enhance the performance of detection, post processing is added. Then the data in “DIC-C2DH-HeLa" are post processed using GaussianBlur and H-maxima transforms. The result images are later dilated to create opening images. Contours are found based on the dilated images. For data in “PhC-C2DL-PSC” , a threshold method is applied. An opening image is then created based on the threshold images. Watershed is applied to the opening images for component analysis.

3.2 Cell detection

Although cell segmentation is completed, segmentation errors still existed. Thus, it is necessary to apply additional evaluation in cell detection.

The developed detected way is to use the opencv build-in function ‘drawcontour’ to detect contour of each cell first. Then, three parameters are set to control the selection range of the contour to improve the performance detection. These thresholds are radius, area and detection factor respectively. The thresholds for area and radius are used to select cells within a specified size range, which depends on the cell size of each dataset. Detection factor is a similarity threshold used to find mitotic cells. As seen from Figure 3, we calculated the convex contour with red line and concave contours with green line respectively for each cell, then using template matching to calculate the similarity between these two kinds of contour shapes. By observing the performance of the division cell in each frame to decide the threshold of detection factor for each dataset. If the similarity of shape matching is greater than setted threshold, that cell will be detected as a mitosis.

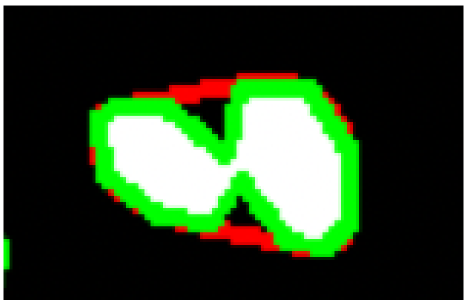


figure 3. Sample image of mitosis detection.

Therefore, each different data set has a corresponding threshold obtained by manual tuning. However, due to time limits, this detected mitosis method does not perform well on the dataset1 because most mitosis cells have brighter spots. In the light of this, the Multi-Otsu Thresholding would be able to be considered to improve detection performance of cell mitosis in dataset1[8].

Once cell detection has been finished, the number of cells in each frame can be calculated. It also has a structure to record information of each detected cell such as center location, cell id, radius, area and frame. These are important factors for cell localization in the nexts cell tracking part.

3.3 Cell tracking

The cell tracking is implemented by Lucas-Kanade optical flow algorithm. Basically it tracks a list of points in the current frame and returns a list of corresponding points in the next frame. According to the official documents of OpenCV[6]: "Optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of an object or camera. It is a 2D vector field where each vector is a displacement vector showing the movement of points from first frame to second." If a pixel I(x, y, t) in first frame moves in the direction of (d_X, d_Y) in next frame in the time of dt, the position of the same pixel in the next frame can be represented by
I(x, y, t)=I($x+d_X, y+dy, t+dt$).
After taking Taylor series approximation of right-hand side, and divide both sides by d_t. The equation becomes like the following:

$$f_x u + f_y v + f_t = 0$$

$$f_x = df/dx$$

$$f_y = df/dy$$

$$u = dx/dt$$

$$v = dy/dt$$

However, while the f_x and f_y can be solved, the u and v are unknown. The Lucas-Kanade method is later introduced to take 9 pixels around the selected pixel to form 9 equations with two unknown variables.

For the tracking processing, the top priority is calculating the center point of each cell, which will be taken as the representative point for every single cell. After the cells are well segmented, the method we used to calculate the center point of each cell is to find the minimal enclosed circle, fitting it into each cell. This means the cell can be approximated by the circle shape. Then from the geometric position of each circle, the center can be calculated. These cell centers will then be converted to a vector of 2D points, passed into the optical flow function along with segmentation result of current image frame and the segmentation result of next image frame. This function calculates an optical flow for a center point set using the iterative Lucas-Kanade method with pyramids. The window size of Lucus-Kanade parameter is set to be 15 x15 pixels in order to handle the irregular movement and relative far motion of cells in the image set Fluo-N2DL-HeLa. On the other hand, since cells in PhC-C2DL-PSC have a smaller area and have stable movements over frames, the window size is reasonably changed to be 10 x10 pixels, which achieves a more accurate tracking performance. To visualise the cell tracking, a line is drawn to connect two corresponding cells in consecutive frames to illustrate the motion of the cell.

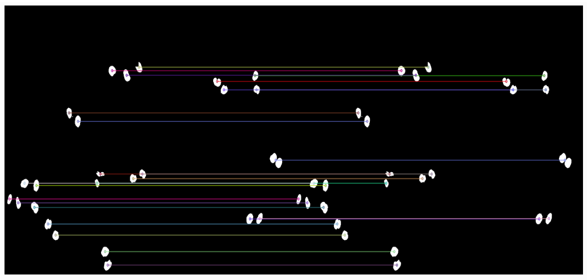


figure 2 cell matching when tracking cells

3.4 Analysis of cellular motion

Four different properties of cellular motion are analyzed in real time for a single selected cell. They include the total distance, net distance, speed and confinement rate of the cell. The tracking algorithm is employed to trace cell frame by frame. For each frame, the motion data of the cell is re-calculated based on the data that was computed in previous frames. The distance measurement between two positions is the Euclidean metric.

The position of the cell is the coordinate of its center. The position of the cell in original frame is denoted as $Position_{original}$, the position of the cell current frame as $Position_{current}$, the total distance as $Distance_{total}$, the net distance as $Distance_{net}$, the speed as $Speed$ and the confinement ratio as $Confinement_{ratio}$.

The total distance of the cell is calculated by the sum of the distance traveled by cell from last frame to current frame and the previous total distance.

$$Distance_{new} = ||Position_{current} - Position_{last}||$$

$$Distance_{total} = Distance_{PrevTotal} + Distance_{new}$$

The net distance of the cell is simply the Euclidean distance between the cell position in current frame and original cell position.

$$Distance_{net} = ||Position_{current} - Position_{original}||$$

The speed of the cell is the distance difference between current position and previous position divided by the number of frames in between.

$$Speed = ||Position_{current} - Position_{prev}|| / Frame\#$$

The confinement ratio is the ratio between the total distance and the net distance traveled by the cell.

$$Confinement_{ratio} = Distance_{total} = Distance_{total} \; / \; Distance_{net}$$

Overall, all of these four parameters provide representative real-time information for a selected cell, which gives a better analysis of the cell from a statistical perspective.

4. Experimental Setup

4.1 Experimental setup

In order to test the performance of the proposed method, we conduct a series of experiments on the given image sequences. The specific information about the dataset are listed in table 1. The experiments are performed on an Intel i7 2.7GHz processor with 16GB of RAM. The code for this project is written in Python 3.7 and requires the following packages installed, including opencv 3.4.2, tensorflow 2.3.0, numpy, skimage, scipy. There is also a readme.txt in the code files that explains how to run the code.

Dataset	Image size	Microscopy imaging technique
DIC-C2DH-HeLa	512*512	Differential Interference Contrast
Fluo-N2DL-HeLa	1100*700	Fluorescence
PhC-C2DL-PSC	720*576	Phase Contrast

table 1

4.2 Evaluation method

We evaluate the performance for the detection model as well as the tracking model. Each part is presented both qualitatively and quantitatively.

For evaluation of detection, we first compare the result of our proposed method of different datasets with each other to see the effectiveness of cell segmentation. We also make comparisons between the proposed method and another common method. The quantitative evaluation is based on the similarity of actual cell number and detected number of every frame. We calculate TP, FP, FN and detailed explanations of these three parameters are shown in table 2. We define Presion as the ratio of the number of detected true cells to the total number of detected cells, and Recall as the ratio of the number of detected true cells to the number of actual cells. The corresponding formulas (1) and (2) are listed below.

	the number of actual true cells	the number of actual false cells
the number of detected cells	TP	FP
the number of not detected cells	/	FN

table 2

$$Precision = TP \,/\, (FP + TP) \qquad (1)$$

$$Recall = TP \,/\, (FN + TP) \qquad (2)$$

For evaluation of tracking, we first compare the results of some representative frames of various cellular density as a qualitative analysis. For every sequence, we go through every cell of every frame to check if it is successfully tracked on the next frame. In order to determine the traceability of the proposed tracking measure, we set various step sizes to see how many cells can be tracked within such a step. For example, suppose there are 13 frames of a sequence and the step size is set as 10, let a[1] denote the ratio of the number of correctly tracked cells of frame 1 in frame 10 to the total number of cells of frame 1. Similarly, we obtain a[2] and a[3], which separately cover frame 2 to frame 11 and frame 3 to frame 12. The final mean accuracy is the average value of a[1], a[2], a[3].

5.Results and Discussion

Results

To demonstrate the performance of the proposed method, all three datasets, “DIC-C2DH-HeLa”, “Fluo-N2DL-HeLa”, “PhC-C2DL-PSC” have been tested. The images in all of the three datasets have low contrast, especially in “DIC-C2DH-HeLa”. In addition, a majority of the images have touching and even overlapping cells.

Sample images for cell segmentation:

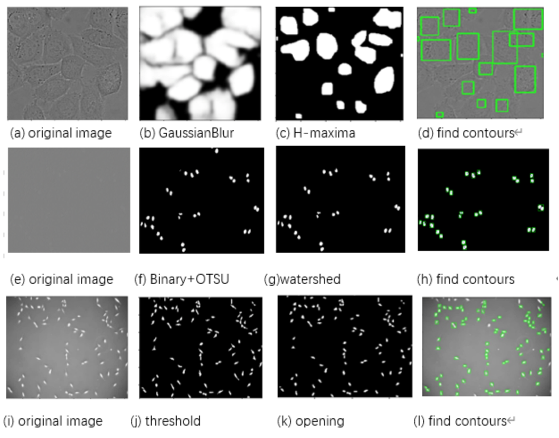


figure 3 (a,b,c,d are from dataset DIC-C2DH-HeLa, e,f,g,h are from dataset Fluo-N2DL-HeLa and i,j,k,l are from dataset PhC-C2DL-PSC)

Sample images for cell tracking:

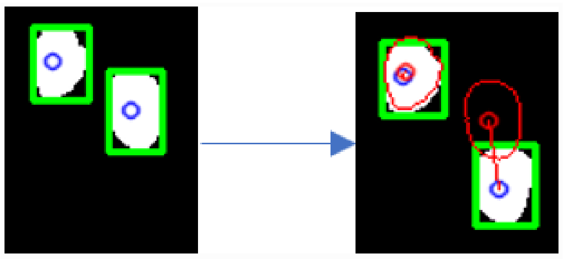


figure 4 (The red line represents the movements of the center of the cells between this frame and the last frame. The first image represents the previous frame of the cells. And the second image shows the contour of the previous frame on the current frame.)

Sample image for cell division:

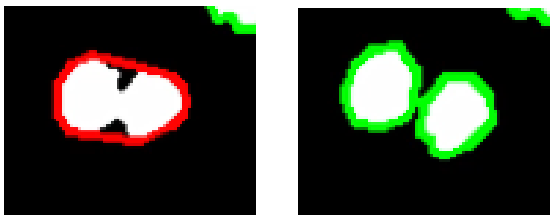


figure 5 (The red line represents the cell that is undergoing a division, the next frame is the two separated cells captured by green circle)

A comparison of binarized images obtained by different methods

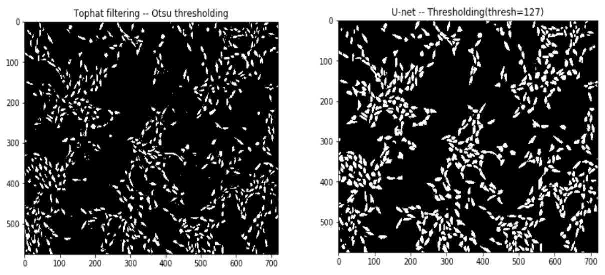
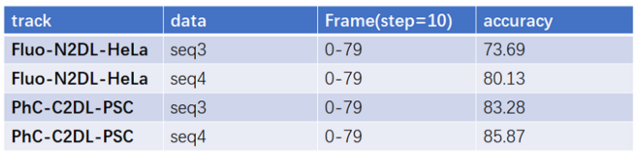

Tophat filtering -- Otsu thresholding
U-net -- Thresholding(thresh=127)

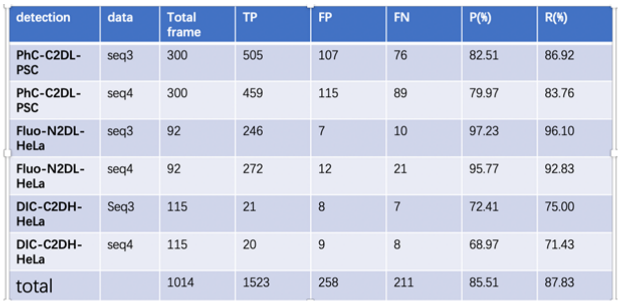
figure 6 (the first image used Top Hat filtering --Otsu Thresholding. the second one used the U-net --Thresholding)

Quantative results:



track	data	Frame(step=10)	accuracy
Fluo-N2DL-HeLa	seq3	0-79	73.69
Fluo-N2DL-HeLa	seq4	0-79	80.13
PhC-C2DL-PSC	seq3	0-79	83.28
PhC-C2DL-PSC	seq4	0-79	85.87

figure 7. Accuracy of correctly tracked cells



detection	data	Total frame	TP	FP	FN	P(%)	R(%)
PhC-C2DL-PSC	seq3	300	505	107	76	82.51	86.92
PhC-C2DL-PSC	seq4	300	459	115	89	79.97	83.76
Fluo-N2DL-HeLa	seq3	92	246	7	10	97.23	96.10
Fluo-N2DL-HeLa	seq4	92	272	12	21	95.77	92.83
DIC-C2DH-HeLa	Seq3	115	21	8	7	72.41	75.00
DIC-C2DH-HeLa	seq4	115	20	9	8	68.97	71.43
total		1014	1523	258	211	85.51	87.83

figure 8. Accuracy of correctly detected cells

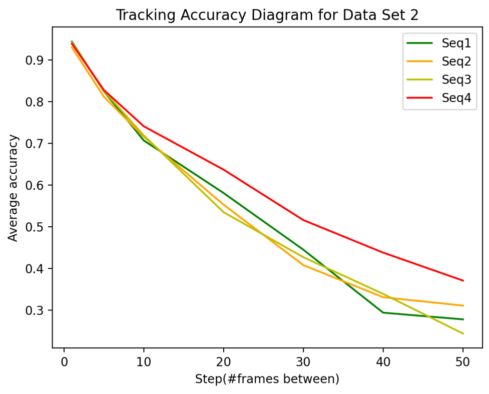


figure 9. Tracking Accuracy for Fluo-N2DL-HeLa

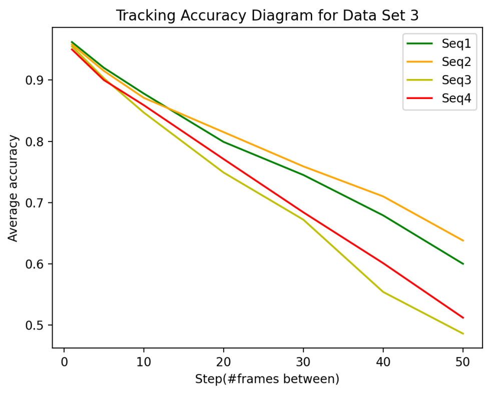
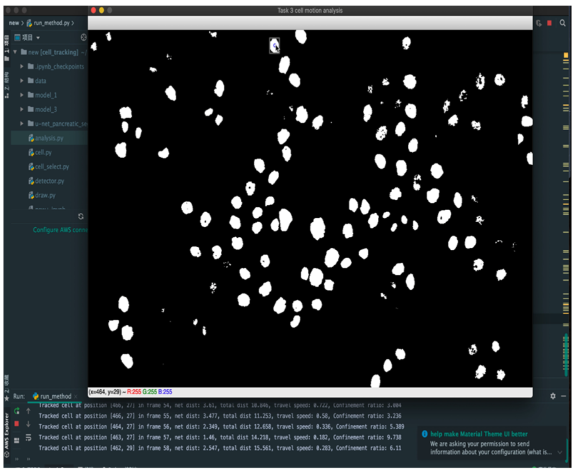


figure 10. Tracking Accuracy for PhC-C2DL-PSC



```
Tracked cell at position [466, 27] in frame 54, net dist: 3.61, total dist 10.846, travel speed: 0.722, Confinement ratio: 3.004
Tracked cell at position [466, 27] in frame 55, net dist: 3.477, total dist 11.253, travel speed: 0.58, Confinement ratio: 3.236
Tracked cell at position [464, 27] in frame 56, net dist: 2.349, total dist 12.658, travel speed: 0.336, Confinement ratio: 5.389
Tracked cell at position [463, 27] in frame 57, net dist: 1.46, total dist 14.218, travel speed: 0.182, Confinement ratio: 9.738
Tracked cell at position [462, 29] in frame 58, net dist: 2.547, total dist 15.561, travel speed: 0.283, Confinement ratio: 6.11
```

Task 3 cell motion analysis
run_method.py
(x=464, y=29) ~ R:255 G:255 B:255
Run: run_method
help make Material Theme UI better
We are asking your permission to send information about your configuration (what is...

figure 11. Sample image of cell motion analysis.

Discussion

Aiming at tracing biological cells in time-lapse microscopy images, optical flow is used to track a list of corresponding points in the next frame. The record of the contour, which comes from the previous frame can show the change of cells on the image of the next frame. This has already been shown in figure 4. Additionally, the combination of U-net, GaussianBlurBlur and h-maximum transform is suitable for the dataset “DIC-C2DH-HeLa”. Because the structure of this dataset is different from another two datasets. The use of U-net can help ensure that the boundary points of the image can be retained to the last layer of the feature map. Moreover, U-net can reduce noise points effectively. As shown on the figure 3, b and c can show the cells’ contour clearly.

Disadvantages:

With the number of frames increases, the tracking accuracy is decreasing due to the irregular cell movement, large number of cell mitosis, and segmentation error. A large amount of the cells are doing mitosis. That will make the tracking more difficult. In addition, optical flow is not the most suitable method for cell tracking because optical flow only tracks objects with a consistent shape. However, cell division will dramatically change the shape of the original cell and create untracked cells in the next frames.

There are also some cells that have under segmentation and over segmentation error in the resulting images observed. Especially in the dataset “DIC-C2DH-HeLa” which contains some touching and even overlapping cells and mitosis. These increase the errors of segmentation.

6. Conclusion

The project proposed a hybrid method for cell segmentation and tracking which achieved relatively satisfying results. The methods use two paths for different datasets to produce the best possible results. The segmentation module applies U-net for “DIC-C2DH-HeLa” and “PhC-C2DL-PSC” to predict the foreground pixels. Those predictions are then fed to the post processing stage. The images in the set “DIC-C2DH-HeLa” are then processed with GaussianBlur and H-maxima. Images in “PhC-C2DL-PSC” are processed with thresholding and watershed. While the “Fluo-N2DL-HeLa” is processed using a top-hat filter and watershed methods.

After the cells have been successfully segmented, cells are localized and calculated for centers which will be used for tracking and mitosis detection. Optical flow is the main method used for tracking, morphological matching is used for detection for mitosis. The proposed method has shown good results even in complex images.

In conducting the experiments, drops in accuracy is noticed when the frame number is increasing. This may be due to the optical flow method failing to track cells that are undergoing mitosis and irregular shapes of the cells. Time permitting, other methods will be considered to track the cells,

In conclusion, the proposed methods achieved the expected outcome but there is room for improvements.

7.Contributions of Group Members

Every member of our group has been working hard for this project and the allocation of tasks and contributions are quite equal and reasonable.

Shuonan Wang (z5158229) implemented cell detection and tracking in Task1, and cell division detection in Task2. He also integrated code from all team members, given data results and video demo.

Zhihan Qin (z5290141) was responsible for the researching and algorithm parts of cell detection for dataset 1 and 3, as well as the evaluation method. She introduced a segmentation method combining the traditional image processing method with the deep learning method which has proved to improve the performance.

Haoran Wang (z5141959) worked on the cell tracking part for image data set 2 and 3 by implementing the optical flow method and then applying it for tracing cell motion. He also calculated the cell motion data in task3 and did a method for visualizing cell motion.

Yukang Yan (z5158298) was in charge of cell segmentation and noise reduction for tracking in task1. He also implemented evaluation methods for all three datasets.

Yuanfang Zhang (z5142254) was in charge of report writing and data-collection. He also did manual annotation for images which were eventually not used due to the change in the selection of deep learning methods. He was also in charge of demonstration and presentation as well.

8. Reference

[1] 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.https://doi.org/10.1016/j.semcdb.2009.07.004

[2] E. Meijering et al. Methods for cell and particle tracking. Methods in Enzymology, vol. 504, no. 9, pp. 183- 200, February 2012. https://doi.org/10.1016/B978-0-12-391857-4.00009-4

[3] Y. Li et al. Detection and tracking of overlapping cell nuclei for large scale mitosis analyses. BMC
Bioinformatics, vol. 17, no. 1, p. 183, April 2016.
https://doi.org/10.1186/s12859-016-1030-9

[4] T. Falk et al. U-Net: deep learning for cell counting, detection, and morphometry. Nature Methods, vol. 16,no. 1, pp. 67-70, January 2019. https://doi.org/10.1038/s41592-018-0261-2

[5] Lux, F., & Matula, P. (2020). Cell Segmentation by Combining Marker-Controlled Watershed and Deep Learning. arXiv preprint arXiv:2004.01607.

[6]"OpenCV:OpticalFlow", Docs.opencv.org, 2020. [Online]. Available: https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html. [Accessed: 02- Aug- 2020].

[7]deepimagej/python4deepimagej", GitHub, 2020. [Online]. Available: https://github.com/deepimagej/python4deepimagej/tree/master/unet/data. [Accessed: 05- Aug- 2020].

[8]."Segment human cells (in mitosis) — skimage v0.18.dev0 docs", Scikit-image.org, 2020. [Online]. Available:
https://scikit-image.org/docs/dev/auto_examples/applications/plot_human_mitosis.html#sphx-glr-auto-examples-applications-plot-human-mitosis-py.
[Accessed: 07- Aug- 2020].