COMP9517: Computer Vision Project Report

Rafael Formoso, Sebastian Castillo Castro, Xinli Wang, Yuzhong Duan, Kan-Lin Lu

Abstract – Detection and tracking of cells plays a vital role in cellular level researches. As part of COMP9517 studies, a team-developed method is produced and presented, consists of UNET segmentation follow by post processing in fine tuning. Tracking and Division are achieved by distance-based approach. Outcome of the proposed method showed reasonable accuracies across given datasets. Limitation of the method mainly involves in over segmentation and single-feature based tracking.

I. INTRODUCTION

Cell detection and tracking through computer vision algorithms have gain increasing focus in recent years. The challenges faced can be categorized into two categories: spatial and temporal[1], summarized as the following:

- Spatial:
  Influence due to image acquisition technique, described as noise. Inconsistent cell area, including deformation, overlapping and irregular shape.
- Temporal:
  Unpredictable cell behavior, including topology change caused by cell life cycle, uneven movement and overlapping across frames.

These challenges become acute when large cell population exists within a single frame. Complete solution requires a robust and self-adjustable algorithm in tackling across different cell of interests, and will revolutionize the current practices in cell level studies, considering computers provide higher level of accuracy and reliability.

As part of computer vision studies for COMP9715, a project is laid out where three different datasets are given, each with four different sequences. The motivation of this project is to implemented learnt computer vision techniques, but not limited to, in providing a reasonable outcome with solid ground evaluation in cell detection and tracking within 5 weeks period. Open source programs are allowed, however the team goal is developing own algorithm in tackling each task. Certain sequence images are provided with masks for machine learning algorithm training purposes.

Following provides a description of given datasets and detail break down of task specifications. Figure 1 presents respective dataset example.

Dataset 1 – DIC-C2DH-HeLa

DIC-C2DH-HeLa, Figure 1 (Left), has a dense layout of HeLa cells where the neighboring cells are often closely connected with each other making it potentially difficult for any computer vision methods to accurately segment the cells or identify any cells that are undergoing mitosis events. The objects of interest in this dataset are the Hela cells, and comparing to other datasets, containing larger cell area. The sequence of images is obtained by label-free optical microscopy technique called Differential Interference Contrast (DIC) of live cells and organisms' cellular structure in 2D high resolution.

Dataset 2 – Fluo-N2DL-HeLa

Fluo-N2DL-HeLa, Figure 1 (Mid), presents a relative higher density comparing to DIC-C2DH-HeLa, with a wide range of intensities between cells and backgrounds and low spatial resolution. The objects of interest in this dataset are the nuclei of HeLa cells, however it is very clear that it will be challenging for human eyes to accurately identify the cells and the presence of any mitosis events without processing the raw images. The sequence of images is captured using fluorescence (widefield, confocal, light sheet) microscopy recordings of the of live cells and organisms' nuclear structure in 2D low resolution.

Dataset 3 – PhC-C2DL-PSC

PhC-C2DL-PSC, Figure 1 (Right), has the highest cell densities within a single frame, hence expecting higher computational cost and finer segmentation process, making the cell tracking exercise a difficult task. This set of images represent Pancreatic stem cells and these are the main object of interest. The sequence of images is captured using phase contrast enhancing of lives cells and organisms' cellular structure in 2D low resolution.

Figure 1 : (Left) DIC-C2DH-HeLa, (Mid) Fluo-N2DL-HeLa, (Right) PhC-C2DL-PSC

Task 1: Detect and Track Cells

The task requires to detect and track cells within given sequences, including establishing techniques to accurately detect cells. This involves, draw bounding box around each cell for the original image, draw the trajectory of each cell, including past trajectory up to a certain time point. Finally, print out a real-time count of the cells detected in each image of the sequence.

Task 2: Detect Cell Divisions

The task requires to expand on top of *Task 1* and further detecting mitosis event, drawing a unique bounding box for cell undergoing division. The daughter cells from division will use either different colour bounding box in order to provide visual inspection. Finally, print out a real-time count of the cells that are undergoing mitosis during any time point.

Task 3: Analyze Cell Motion

The task requires to build a program that could analyze and extract the motion of any selected cells. Speed of the cell at any time point can be estimated by computing the Euclidean distance between the coordinates of the cell's bounding box center in the consecutive frames divided by the time difference. Compute the total distances travelled by any cell up to an arbitrary time point using the cumulative Euclidean distances from all the past frames. Net distance travelled by any cell from the first time point of its trajectory to the current time point. Confinement ratio of the cell mention as Total distance / Net distance.

This report is structured as the following. Section II provides a brief literature review on related work and background understanding, including cell behaviors. Section III provides a justification and explanation of selected techniques. Section IV explains the setup of the experiment along with evaluation methods that were adopted. Section V provides statistical and visual results for proposed method, discussion of performance and outcome is also provided. Section VI concludes the report with recommendation for future work. Lastly, Section VII provides each group member's contribution towards the overall project.

II. Literature Review

This section provides a brief literature review conducted for background and theory understanding.

It was discussed in [2] that tracking algorithms can categorized into tracking by segmentation, by model evaluation and by filtering. The task of segmentation often involves identifying individual foreground objects in a single frame and, in this project, individual cells. Tracking is then performed by identifying the same cell in subsequent frames, utilizing the cell position within the frame overtime.

Traditional segmentation methods such as thresholding, template matching, watershed transformation and deformable models were discussed in [1]. In recent studies, thresholding is often considered one of the most popular segmentation approaches, but it is also the most error prone. [1] found the thresholding tend to perform better when the cells are generally well separated, especially when the cell intensities differ sufficiently from the background on a consistent basis. Template matching works well for images showing consistent cell shape but often performs badly in the case of significant variations in cell morphology either between cells per image or per cell over time or both. Watersheds transform often requires time consuming pre and post processing techniques to be performed and one of the main drawbacks of this method is due to its sensitivity to noise in images. This tendency often leads to over segmentation.

In [2] a range of cell division detection methods were discussed. Due to the limitations of tracking based and tracking free method, more hybrid approaches were developed that takes advantage of the visual and temporal dynamic information of cells potentially undergoing mitosis. Hybrid approach applies graphical models on candidate image sequences extracted short-term tracking [2].

Deep learning-based methods performs better over traditional methods on mitosis detection, this is mainly due to the ability of deep learning approach been able to extract and learn much more data relevant and discriminative features from the given dataset. Limitations of deep learning approach includes the network often needs to be sufficiently complex to be able to describe the visual and dynamic features of deformable cells [2]. However, it is widely known that the scale of the datasets in biomedical imaging is limited and often requires a higher level of expert knowledge to accurately annotate cells. This manual annotation process is often time consuming and error prone. It is also worth mentioning the deep learning models trained on these annotated masks will have the weakness of overfitting.

Cell motion analysis was discussed in depth in [1]. One of the approaches discussed in the paper was to associate each cell in any frame to the spatially nearest cell in the frame using the cell's centroid position [1]. The biggest drawback with this approach is dealing with dense cell populations or rapid cell movements, this will often lead to mismatches, however it is worth mentioning that the described approach is simple to implement and worth considering given the short amount of time. The proposed method in the paper utilized cell features such as similarity in cell intensity, the area of the cell, orientation of the cells, boundary curvature and estimated displacement.

Cell motion measures such as speed, direction, net displacement, max displacement, and mean-squared displacement (MSD) were also discussed in detail in [3]. It was highlighted that cells were often known not to move with constant speed or in the same direction, so measures such as MSD could be utilized to quantify whether the cell of interest is moving randomly (Brownian motion) or in a more directed fashion following certain signals.

III. METHODS

This section provides a breakdown of selected technique for each dataset, along with justification.

A. Task 1: Detect and Track Cells

U-Net

We note that a recurring weakness of classical techniques on this dataset is their inability to properly separate different cells, leading to incorrect results when we attempt to run the watershed algorithm. In addition, cells in datasets such as the DIC-C2DH-HeLa dataset exhibit a wide variety of textures, making it difficult to develop a generalized approach to segmenting them. To address these issues, we opted to run these images through U-Net, [4], noting its ability to deliberately penalize misclassified cell borders.

We trained a custom variant of U-Net on each dataset separately. Our variant of U-Net includes a scaling parameter for the number of channels in each layer and is trained to simultaneously perform semantic segmentation while providing an estimate of the distance transform given each cell. Like the original U-Net, we classify borders between touching cells as part of the background, penalizing misclassified border pixels by a factor of 1.5. To obtain the cell borders, we took each individual cell detection from the ground truth data and performed dilation, marking overlapped dilations as borders.

For our implementation, we used a model scaling parameter of 0.6, with a learning rate of 0.0007 using the Adam optimizer, and trained with a batch size of 3. We used binary cross entropy as the segmentation mask's loss function and mean squared error as the distance estimator's loss function. For data augmentation, we used random cropping, flipping, rotation, and skew adjustment.

Dataset 1 – DIC-C2DH-HeLa Detection

To emphasize each image's features prior to running it through U-Net, we performed Contrast Limited Adaptive Histogram Equalization (CLAHE), with the clip limit set to 4.0. We then trained U-Net on this dataset for 1500 iterations.

Post-Processing of DIC-C2DH-HeLa after UNET output mainly involves watershed segmentation using given distance image. As observed from Figure 2 Row 1, tiny noises still present after UNET segmentation(Mid), watershed output would consist mislabeled segments, hence in order to filter out tiny noises, reconstruction process is taken place after watershed using segmented image, using the following equation, provided in lecture note:

$$\boldsymbol{R_i = (R_{i-1} \oplus S) \cap I \quad Until\ R_i = R_{i-1}} \qquad ①$$

Centroid can then be located via built-in findContours function within OpenCv[5] utilizing reconstruct-segmented image. However, comparing to other datasets, DIC-C2DH-HeLa image consists of several inner segmentation 'holes' that causes mislabeling, hence RETR_EXTERNAL is utilized to return only extreme outer flags[6]. Morphological operation involving simple erosion and dilation is also conducted on reconstruct-segmented image in order to clearly outline different cells using kernels with various shapes and sizes, when large overlap occurs.

In order to justify choice of watershed segmentation, comparison is made against method using morphological

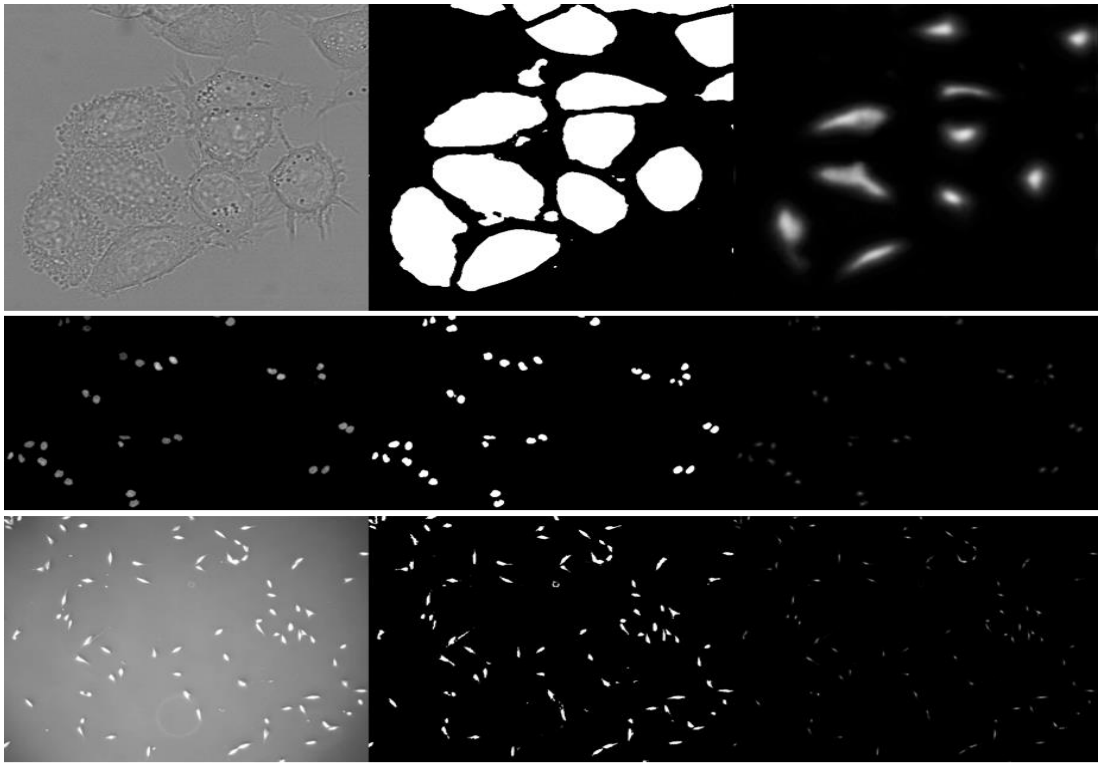

Figure 2 : Neural Network Outcome illustration, (Row-1) DIC-C2DH-HeLa, (Row-2) Fluo-N2DL-HeLa, (Row-3) PhC-C2DL-PSC.

operation (ultimate erosion and reconstruction: I - D - M - R), whilst such approach performs well under single image operation, cross frame and cross sequences outcomes showed over-segmentation consequence, considering the irregular size of given cell shape. Other methodologies, including laboratory exercises method, such as mean-shift, were also performed, however result is not ideal with connected cell.

Dataset 2 – Fluo-N2DL-HeLa Detection

For the Fluo-N2DL-HeLa dataset, we increased the contrast of the image so the darkest points would have a value of 0, while the brightest points would have a value of 1. We then applied a mask, setting all values lower than the mean of the image to zero. The image is then upscaled to 1.5 times its original size. We then trained U-Net on this dataset for 750 iterations.

Post-Processing of Fluo-N2DL-HeLa mainly involving similar process comparing to DIC-C2DH-HeLa, however considering the average size of cells presented, see Figure 2 Row 2, reconstruction is not performed as outcome of watershed segmentation provides distinguishable cell areas. Single erosion iteration is conducted in order to clearly distinguish larger connected cells for centroid labelling. Comparing to DIC-C2DH-HeLa, through visual inspection, no within cell inner segmentation 'holes' were observed, hence under the same built-in findContours function, RETER_TREE hierarchy is utilized, where the function retrieves all contours[6].

In order to justify choices of segmentation method under Fluo-N2DL-HeLa, similar comparison was made against method using morphological operation and other taught approaches, however over segmented outcome is observed, where cross frame and cross sequence result is not as ideal comparing to UNET and post processing approach.

Dataset 3 – PhC-C2DL-PSC Detection

For the PhC-C2DL-PSC dataset, we first performed background subtraction by subtracting the image's morphological opening from itself, with its kernel size set to 15. We then performed the same contrast stretching and filtering effects as those applied to the Fluo-N2DL-HeLa dataset. We then trained U-Net on this dataset for 100 iterations, noting its quick convergence due to the limited complexity of the data.

Post-Processing of PhC-C2DL-PSC does not involve watershed segmentation, as outcome from UNET showed clear distinction of cells, see Figure 2 Row 3. Centroid were located using findContours function under RETER_TREE hierarchy.

In order to justify choices of segmentation method under PhC-C2DL-PSC, comparison was made against manual segmentation and watershed methods. The former method present similar consequence as mentioned in DIC-C2DH-HeLa and Fluo-N2DL-HeLa. The latter provides a more interesting comparison, where considering large overlapping, high population and on-spot spawning of cells behaviors observed in PhC-C2DL-PSC cells, watershed method did not provide a better outcome comparing to raw segment image from UNET, at the same time having higher computational cost, hence disregarded.

Tracking Cells

Tracking cell requires mainly finding matching cells between 2 consecutive frames. The process mainly involving a distance cost matrix operation, shown in Table 1.

Given the list of centroids from previous image and current image, the matrix is formed where the column represents the cells within current image, and row represents the cells from previous image. Distance is calculated using Euclidean distance equation, shown in the following:

$$\boldsymbol{d} = \sqrt{(\boldsymbol{x_1} - \boldsymbol{x_2})^2 + (\boldsymbol{y_1} - \boldsymbol{y_2})^2} \quad ②$$

Once all values within each dataframe cell is calculated, column operation is first performed by find the column minimum observed, which represents the current cell's closest pair within previous image, see ID 3 of current image in Table 1, where column minimum has a value of 1 pixel unit with ID 3 of previous image. This also happens to be the row minimum, in other words, previous cell's closest pair within current image, see ID 3 of previous image in Table 1. However, column minimum does not necessary represent matching previous cell, as shown in ID 4 of previous image in Table 1, where ID 4 of current image has a much closer distance, see *Task 2: Detect Cell Divisions* section for further explanation. Matched cell will carry on the same Cell ID internally within the program.

Table 1 : Illustration of distance matrix operation, (Green Column) Divided cell locating, (Yellow Column) Matched cell locating.

	Matched	Current Image Cell IDs			
	Divided	1	2	3	4
Previous Image Cell IDs	1	32	12	2	15
	2	3	17	3	41
	3	35	4	1	11
	4	4	12	84	1
	5	21	34	9	5
	6	31	7	12	3

In order to provide a complete trace of overall trajectory a cell dataframe is kept in recording centroid location at respective time frame, outputted as html file under Results-Annotation folder, example is shown in Figure 3. Trajectory drawing is done in two ways: simple previous location drawing (represented in blue line on output image) and 3D pathology graph, along with Total Cell Number count per frame – both shown in Figure 4.

	centroid_list	zlist	current_z	current_centroid	task_3_1	task_3_2	task_3_3	task_3_4
id								
0	[(208, 486)]	[0]	0	(208, 486)	0	0	0	Division Zero
1	[(339, 433)]	[0]	0	(339, 433)	0	0	0	Division Zero
2	[(412, 361)]	[0]	0	(412, 361)	0	0	0	Division Zero

Figure 3 : Dataframe illustration.

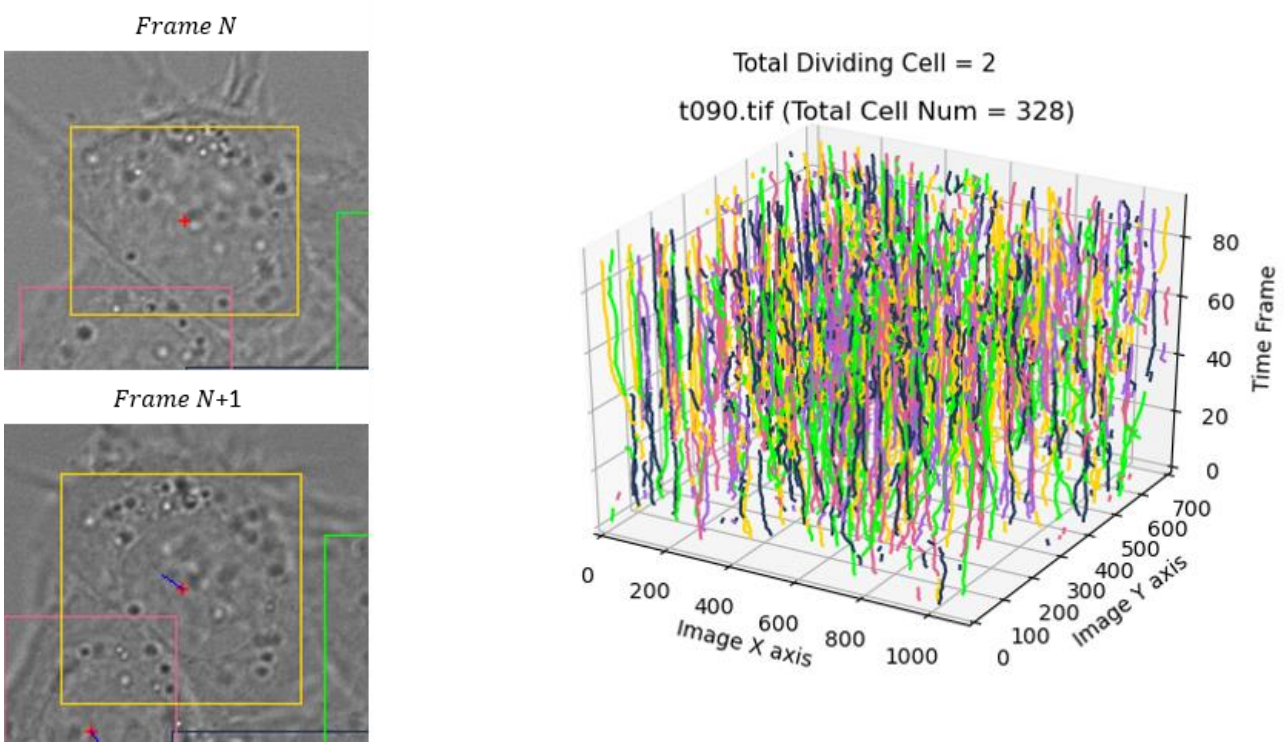

Figure 4 : (Left) Simple previous locations trajectory using DIC-C2DH-HeLa, (Right) 3D Pathology graph using Fluo-N2DL-HeLa. 3D pathology graph can select a single cell for drawing, see *Task 3: Analyse Cell Motion* section.

B. Task 2: Detect Cell Divisions

Mitosis, or cell division is detected via a defined distance, δ, this is based on an assumption made from background understanding shown in previous sections, consultation with academic tutor and group collective discussion. Considering the mitosis behavior shown in Figure 5 (Left), during the process there is an increase in total area of two connected circular areas, after the process, two extremely close centroids are created relative to previous single centroid location. This behavior is also observed within given datasets, specifically shown in DIC-C2DH-HeLa, an illustration is shown in Figure 9.

From this observation, a robust distance-based approach is established and shown in Figure 5 (Right). Using Euclidean distance defined in previous section, considering Frame I, with a centroid location, in this scenario Cell-ID 4 is selected. In the next Frame, I+1, a circle with radius δ is considered using Frame I previous centroid location, new centroids that are presented in Frame I+1 within this δ circle is considered as a divided cell from original cell.

The practical implementation of this approach is illustrated in Table 1, using Cell ID 1 in current image as an example. The column minimum based on calculated distance is 4 pixels, however, Cell ID 4 in previous image has a matched closer centroid, Cell ID 4 in current image. Therefore, based on the mentioned above assumption, given a definition of $\delta = 5\ pixels$, conclusion is drawn that Cell ID 1 is a divided cell from Cell ID 4 of previous image. Value of δ varies across dataset and is computed base on trial and error.

The closest divided cell, or daughter cell, will carry on the original Cell's cell id for tracking purpose, and the cell that is further away will have a new marked Cell ID, in this case, current image Cell ID 1 when stored within internal dataframe will have a string value of 4-0, indicating a divided cell from Cell ID 4.

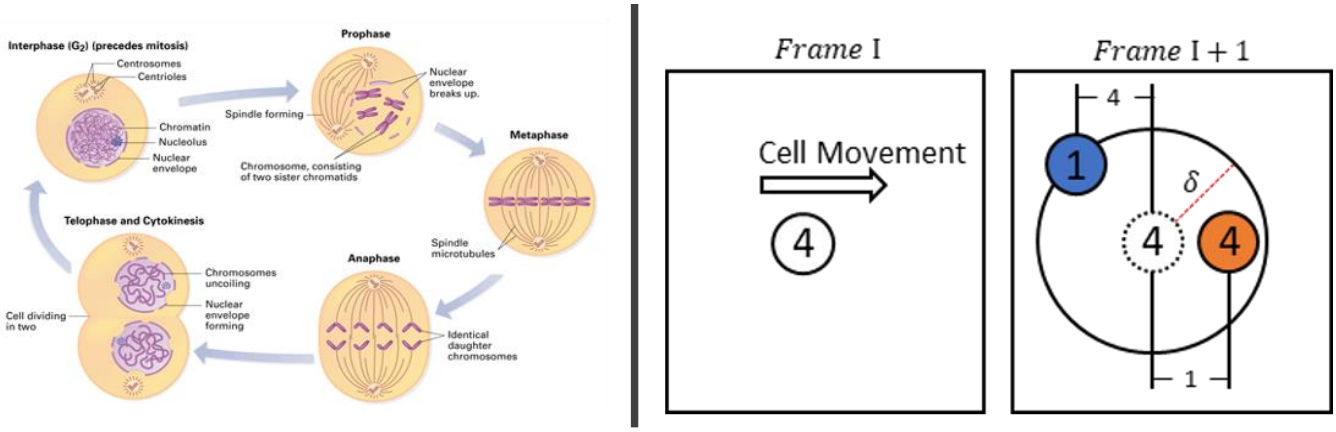

Figure 5 : (Left) Mitosis Process illustration[7]**, (Right) Adopted method for detecting mitosis – not to scale.**

Once identified division, similar to backward patching is utilized, where divided cell will have previous centroid location as initial location, and a highlighted box with red is drawn in marking division took place within previous image, this is illustrated in Figure 9. Total Number of dividing cell per frame is also printed on 3D Pathology graph, shown in Figure 4 (Right), an example of complete trace of a selected cell division is shown in Figure 6.

This extreme robust approach shown relatively good performance, however limitation due to overlapping and segmentation error need to be considered and is discussed in **Results and Discussion** Section, considering the given time duration of the project.

C. Task 3: Analyse Cell Motion

In order to provide the extended program for analysis of the motion of a single selected cell, a cell dataframe is created to store all past centroid locations, as illustrated in Figure 3.

As instructed, speed of cell at specific time frame is estimated by using Euclidean distance divided by time difference. As time difference is always 1 frame, therefore division is always resulting in numerator value, hence the unit of speed is pixels per frame. This is achieved by retrieving the last centroid location from stored list and perform calculation, note for cell that is spawned in the given time frame, a speed of 0 pixel per frame is given, considering the cell has not moved.

Total distance travelled up to specific time frame is provided by the sum of the Euclidean distances computed from initial centroid point to current. This is achieved by iterating over stored list of centroids and summing Euclidean distance between every two points.

Net distance travelled is the Euclidean distance between initial centroid point to current centroid point across time frame. This is achieved by retrieving the initial centroid location and compute distance.

Confinement ratio, as instructed is the simple ratio between the above calculated total distance travelled and net distance travelled. This is achieved by simply retrieving result from previous calculation and perform calculation.

An illustration of overall Task 3 requirements is shown in Figure 6, where t3-1 refers to Task 3-1, t3-2, refers to Task 3-2, t3-3 refers to Task 3-3 and t3-4 refers to Task 3-4. Due to limited width size of figure, units cannot be presented within the output figure. However, all cells presented will have a Task 3 calculation computed within annotation file, under Results-Annotation folder within each respective frame's html file.

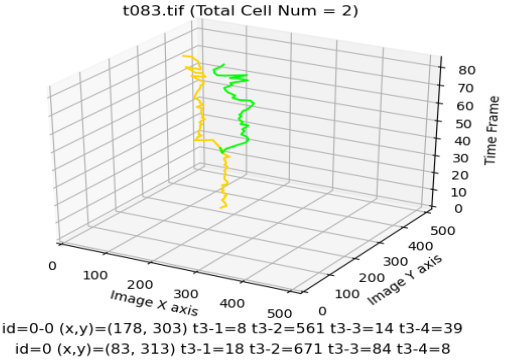

Figure 6 : 3D – Pathology graph, Task 3 Calculation provided as caption.

The implementation of overall Task 3 is completed via user prompt towards the end of *Box_Drawing_Script.py*, where user inputs 'A' for motion Analysis, then input respective cell id retrieved from annotation html file under Results-Annotation directory. Once inputted the desired tracking cell id, the program will re-run the given sequence and specifically look for selected cells and drawing an output 3D pathology graph, along with Task 3 calculation. Instruction is also provided within written script.

IV. EXPERIMENTAL SETUP

This section provides a detailed breakdown of experimental setup and evaluation method conducted for each given task.

A. Task 1: Detect and Track Cells

Detection Evaluation

An exhaustive process was performed consisting of carrying out several cycles of experimentation, where diverse image segmentations were produced in combination with applying different postprocessing techniques (based mainly in a combination of erosion, dilation, closing and opening methods). Subsequent evaluation of the result of each cycle was performed to choose the best possible outcome.

1. Cell count: Comparison between manual cell count and the number of cells detected by the solution, for different sequences (with few, medium and high number of cells)
2. Cell match: For each subset of sequences evaluated in step 1, it was visually checked that bounding boxes were associated with each cell and not with fragments or noise in the image.

Cell count: Figure 7 represents an example where it is possible to check the cell count calculated by the solution at a given moment (see title on top of the 3D graph in Figure 7), and the bounding boxes corresponding to each cell, which shows that the count is being carried out correctly. This is contrasted with a manual count of cells displayed in the corresponding frame.

Cell match: To verify that each detected cell corresponds to an actual cell (and not noise), the visual evaluation was performed on the resulting cell detection image, by verifying each cell present in a specific frame in contrast with the drawn bounding boxes (see Figure 7).

Tracking Evaluation

In order to evaluate the tracking process, a subset of sequences was considered, for which the trajectory was drawn of each cell during the time. The trajectories were checked visually, analyzing them in a 3D graph, where the x and y axes

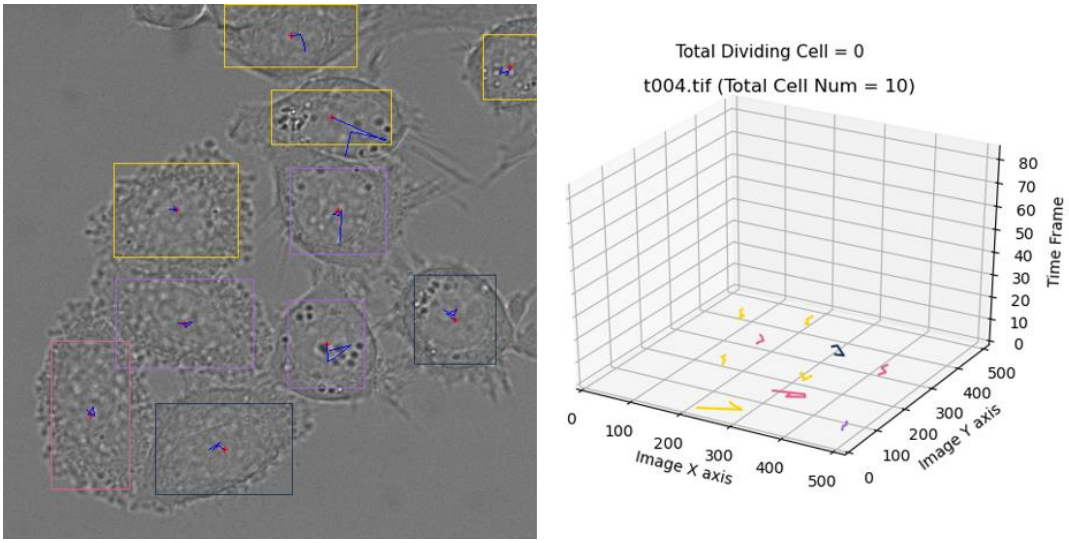

Figure 7 : The number of cells calculated by the solution is shown within the title of 3D Pathology Graph.

represent the coordinates of the cells in the image, and the z axis represents time. Considering that this evaluation becomes complicated for a medium to high number of cells, a trajectory review was also made by plotting the path followed by only one cell, Figure 6.

Every cell in the 3D graph get an assigned color that matches with the color of the corresponding bounding box. Additionally, for each cell in a sequence, a trajectory is drawn with respect to the position of the cell in the previous frames, which is indicated in a blue line, Figure 8.

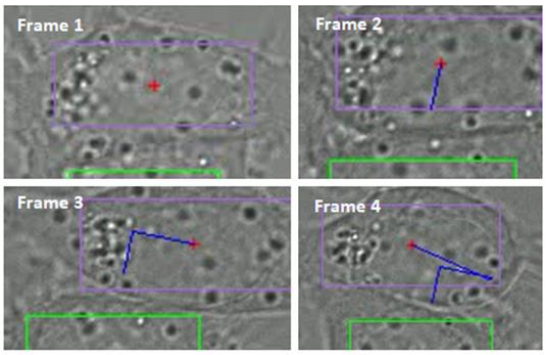

Figure 8 : Trajectory of a single selected cell.

B. Task 2: Detect Cell Divisions

When a mitosis event occurs in frame "N", the cell that is in the process of dividing is highlighted in frame "n-1" (previous frame) by a red bounding box. To validate that mitosis is effectively detected, a visual check was performed on a subset of representative sequences. As an example, Figure 9 presents a screenshot of the moment when mitosis occurred.

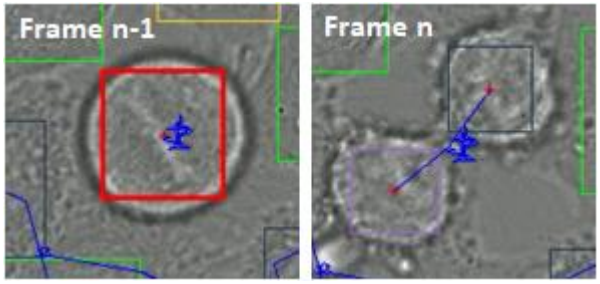

Figure 9 : Mitosis is detected in the frame before it occurs.

A second visual check for detection of mitosis was performed based on the trajectory trace module. In the following image, the path division that occurs at the time of mitosis for a given cell can be seen in Figure 6. In that graph, the x and y axes represent the coordinates of the cell in the image, and the z axis represents time.

C. Task 3: Analyse Cell Motion

By using the tracking modality for only one cell, motion evaluation was performed by checking directly the values calculated in the trajectory module (3D graph) which can be seen in Figure 6. The following information is displayed according with the next labels (units are in pixels):

- Label “t3-1”: Speed of the cell at that time point.
- Label “t3-2”: Total distance travelled up to that time point.t
- Label “t3-3”: Net distance travelled up to that time point
- Label “t3-4”: Confinement ratio of the cell motion.

To evaluate accuracy of the parameters above, they were checked visually frame by frame for a subset of sequences for each dataset. Furthermore, those parameters were summarized for the whole sequences of each Dataset. Parameters like “total number of cells”, “Total division occurrence”, among others, were also calculated.

V. Results and Discussion

This section provides a detailed breakdown of achieved results for each dataset, given the described earlier experimental settings, along with discussion of method performance and outcome of the experiment. A confidence interval (CI) is provided for each calculation for statistical reference, considering the nature of analysis and mentioned earlier possible error, a 90% interval is chosen in indicating spread of data points. Summary of results is provided in Table 3 and Table 4. Due to limited number of pages, refer to output video and respective output image for visual outcome.

Division distance is set up as per Table 2, value is obtained based on trial and error.

Table 2 : Division Distance Setup.

	δ
DIC-C2DH-HeLa	80
Fluo-N2DL-HeLa	10
PhC-C2DH-PSC	3

A. Task 1: Detect and Track Cells

Dataset 1 – DIC-C2DH-HeLa

As observed, average total number of cells presented across all sequences is 99, however this does depend on presented sequence behavior, as some sequence has relative lower cell area but higher population, for instance sequence 3 with total cell number of 165. Tracking of cell is presented within respective annotation folder with list of cell previous centroid locations.

Dataset 2 – Fluo-N2DL-HeLa

As observed, average total number of cells presented across all sequences is 1441, where sequence 2 presents with total cell count of 2768, taking into account the above-mentioned segmentation limitation. Tracking of cell is also presented within respective annotation folder.

Dataset 3 – PhC-C2DL-PSC

As observed, average total number of cells presented across all sequences is 37402, where sequence 1 present the most acute population of 51452 total number of cells presented overall. However, do take into consideration the cross frame over segmentation that might result in mislabeling of new cell or divided cell assignment. This is observed after 300th frame of sequence 1, where cells are located extremely close to one another.

Overall, detection and tracking showed relative acceptable outcome of 97% accuracy by visual inspect, result shown in Table 5, however, over segmentation within PhC-C2DL-PSC does eliminates extremely small cells. In terms of performance, DIC-C2DH-HeLa and Fluo-N2DL-HeLa, take less than 5 minutes on average to perform detection for overall dataset. Comparing to Fluo-N2DL-HeLa, due to the extreme large population, the distance calculation and matching process required higher computational time, for a single

Table 3 : Cross Frame Statistical Summary for DIC-C2DH-HeLa and PhC-C2DL-PSC.

DIC-C2DH-HeLa				
Sequence Number	1	2	3	4
Total Cell Number in Entire sequence	71	96	165	64
Total Cell Number in Last Frame	18	17	9	12
Total Division Occurrence	38	55	67	27
Average Cell Speed (Pixel/Frame)	12.37	14.07	12.96	9.21
Speed 90% CI	(11.77,12.97)	(13.33, 14.82)	(12.25, 13.67)	(8.72, 9.71)
Average Cell Cumulative Distance (Pixel)	213.97	154.23	120.01	190.97
Cumulative Distance 90% CI	(151.15,276.8)	(111.46,197.0)	(90.1, 149.92)	(129.84, 252.11)
Average Cell Net Distance (Pixel):	49.67	46.1	38.09	45.84
Net Distance 90% CI	(42.34, 56.99)	(40.54, 51.66)	(33.04, 43.14)	(36.91, 54.78)
Average Confinement Ratio	5.15	3.46	5.06	5.61
Confinement Ratio 90% CI	(3.5, 6.79)	(2.62, 4.3)	(3.49, 6.62)	(3.32, 7.89)
Fluo-N2DL-HeLa				
Sequence Number	1	2	3	4
Total Cell Number in Entire sequence	770	2768	1020	1207
Total Cell Number in Last Frame	129	334	213	229
Total Division Occurrence	16	112	21	51
Average Cell Speed (Pixel/Frame)	2.4	2.25	2.22	1.9
Speed 90% CI	(2.36, 2.44)	(2.24, 2.28)	(2.19, 2.24)	(1.88, 1.92)
Average Cell Cumulative Distance (Pixel)	23.07	19.43	24.92	24.22
Cumulative Distance 90% CI	(21.04,25.1)	(18.44, 20.43)	(22.96, 26.88)	(22.49, 25.96)
Average Cell Net Distance (Pixel):	6.99	6.56	7.34	6.88
Net Distance 90% CI	(6.46, 7.52)	(6.26, 6.87)	(6.86, 7.82)	(6.47, 7.3)
Average Confinement Ratio	3.33	3.36	3.47	3.85
Confinement Ratio 90% CI	(3.11, 3.55)	(3.18, 3.54)	(3.21, 3.74)	(3.54, 4.15)

Table 4 : Cross Frame Statistical Summary for PhC-C2DL-PSC.

PhC-C2DL-PSC				
Sequence Number	1	2	3	4
Total Cell Number	51452	43864	27369	26923
Total Cell Number in Last Frame	1749	1423	865	795
Total Division Occurrence	1740	1306	519	604
Average Cell Speed (Pixel/Frame)	0.705	0.68	0.66	0.65
Speed 90% CI	(0.7, 0.71)	(0.68, 0.69)	(0.66, 0.66)	(0.65, 0.66)
Average Cell Cumulative Distance (Pixel)	3.73	3.44	2.57	2.41
Cumulative Distance 90% CI	(3.67, 3.78)	(3.38, 3.5)	(2.51,2.63)	(2.35, 2.47)
Average Cell Net Distance (Pixel):	1.44	1.4	1.11	1.02
Net Distance 90% CI	(1.42, 1.46)	(1.38, 1.42)	(1.09,1.14)	(1.01, 1.05)
Average Confinement Ratio	2.53	2.39	2.24	2.28
Confinement Ratio 90% CI	(2.51, 2.57)	(2.36, 2.43)	(2.2, 2.38)	(2.23, 2.33)

sequence takes an average of 40 minutes to finishing performing detection process per sequence.

B. Task 2: Detect Cell Divisions

Dataset 1 – DIC-C2DH-HeLa

As observed, division takes around 48% averaged of total cell number, this to certain extent reflects the spawning of cells within each sequence, however correction needs to be taken into account considering segmentation error and mislabeling due to division distance definition mentioned earlier.

Dataset 2 – Fluo-N2DL-HeLa

As observed, comparing to DIC-C2DH-HeLa, division took place around 3.1% averaged to total cell number, this does reflect within the shown frame as on spot spawning does took place within each sequence.

Dataset 3 – PhC-C2DL-PSC

As observed, on average division took place around 2.6% averaged to total cell number, similarly to Fluo-N2DL-HeLa, this does reflect within given frames, considering on-spot spawning of cells.

Overall, performance of mitosis detection performs relative well under DIC-C2DH-HeLa, with higher cell area. With regard to Fluo-N2DL-HeLa and PhC-C2DL-PSC, as the cell area becomes relatively smaller with higher overlapping, mitosis method undertook begins to have mislabeling, due to cells travelling across each other. Computation of division detection is covered within matching computation, considering distance matrix is generated as a whole.

C. Task 3: Analyse Cell Motion

Average cell speed presented in Table 3 and Table 4 is based on per frame movement, every cell cells' velocity is stored per frame and overall is calculated to provide accurate measurement. Average value for cumulative distance, net distance and confinement ratio are based on overall cell dataframe resulted after completed single sequence run, as this reflects every cells' movement across total life time. Note confinement ratio is not computed based on calculated average cumulative distance and net distance, as such approach does not take into account overall spread of data point.

Dataset 1 – DIC-C2DH-HeLa

A average speed of 12.15 pixel per frame is shown across sequence, respective cumulative and net distance are reported within Table 3. Lowest confinement ratio was observed in Sequence 2, however with highest average speed presented, this reflects the cells within the frames travelled further away from previous centroid – note, it was also observed that sequence 2 presents 57% of division to total cell number, where post-division centroid location tend to be further away from original cell centroid, which contributes to lower confinement ratio, due to increase in net distance.

Dataset 2 – Fluo-N2DL-HeLa

An average speed of 2.28 pixel per frame is shown across sequence, this is reflected in output video, where the cells comparing to DIC-C2DH-HeLa tend to move less. In this aspect, the confinement ratio presented showed slight fluctuation around a value of 3.5 across sequences.

Dataset 3 – PhC-C2DL-PSC

Similar to Fluo-N2DL-HeLa, PhC-C2DL-PSC presents a stable cross sequence average speed of 0.69 pixel per frame. An interesting behavior is observed within Sequence 1, where it presents the highest average cumulative distance travelled with the highest average net distance travelled. After discussion and careful observation across sequence output videos, it is considered due to the extreme large population presented and mentioned above over segmentation error with possible mislabeling of division from overlapping, resulted in such outcome.

Overall motion analysis of cell performs as expected, nevertheless this task is highly depended on *Task 1* and *Task 2*, where mislabeling and over segmentation result in false calculation of final confinement ratio.

Overall Performance Discussion:

The proposed method for cell detection and tracking, with motion analysis show reasonable outcome, considering cross

Table 5 : Accuracy Analysis

DIC-C2DH-HeLa / Sequence 1 (Frame 0 - 14)					
Cell Detection Analysis					
TP	157	FP	4	FN	1
Precision (%)	97.52 %		Recall (%)	97 %	
Cell Division Analysis					
TP	1	FP	3	FN	0
Precision (%)	25.00 %		Recall (%)	80%	
DIC-C2DH-HeLa / Sequence 3 (Frame 0 - 14)					
Cell Detection Analysis					
TP	179	FP	8	FN	0
Precision (%)	95.72%		Recall (%)	100%	
Cell Division Analysis					
TP	2	FP	5	FN	0
Precision (%)	28.57%		Recall (%)	100%	

sequence comparison of results. However, it does present a decrease in reliability when cells are clustered with extreme high population presented within single time frame. Further improvement is required and is discussed in the next section.

Considering the high number of cells present in each dataset, especially in datasets Fluo-N2DL-HeLa and PhC-C2DL-PSC, was unfeasible to do manual checking for each sequence to calculate precisely the performance of cell detection and cell mitosis events. For this reason, dataset DIC-C2DH-HeLa that has relatively low number of cells was used to calculate accuracy for cell detection and cell division. The first 15 frames of Sequence 1 were analyzed manually as a representative subset of the sequence. Based on that analysis, the resulting accuracy for cell detection was 97% and for cell division an 80%. Particularly, for cell division detection, the main errors in the detection, comes from interpreting two cells that get very close to each other as a division. Details are shown in the following table:

VI. Conclusion and Future Work

In conclusion, the proposed self-developed method showed reasonable performance in detection and tracking, mainly involving UNET segmentation with post processing in fine tuning output segmented image, where morphological operation is conducted, and varies across different datasets. Evaluation were mainly conducted via visual inspection. Limitation of the detection method mainly involves in segmentation process, especially for cell that has extremely small area surfaces and having overlapping areas, where over-segmentation or in separable area is shown, hence further investigation in fine-tuning UNET or post processing is required.

As a consequence, tracking and division detection is also influenced by segmentation outcome. However, proposed distance method does also showed decrease in performance when cluster of cells occurred in extremely close distance to one another, specifically within PhC-C2DL-PSC. Hence reliance on single feature for division is not ideal, in future, implementing image-related features in establishing a fine-tuned matching matrix will give better outcome. KD-tree implementation can also be investigated in comparing performance for tacking.

VII. Contribution of Group Members

This section provides a breakdown of team members' contribution towards the project:

Rafael Formoso

- Pre-processing and UNET processes

Sebastian Castillo Castro

- UNET and post-processing

Xinli Wang

- Post-processing and documentation

Yuzhong Duan

- Post-processing and program optimization

Kan-Lin Lu

- Post-processing and documentation

VIII. References

[1] E. Meijering and I. S. , Oleh Dzyubachyk, "Methods for cell and particle tracking," *Methods Enzym.*, vol. 504, pp. 183–200, 2012, doi: 10.1016/B978-0-12-391857-4.00009-4.

[2] A.-A. Liu, Y. Lu, M. Chen, and Y.-T. Su, "Mitosis Detection in Phase Contrast Microscopy Image Sequences of Stem Cell Populations: A Critical Review," *IEEE Trans. Big Data*, 2017, doi: 10.1109/tbdata.2017.2721438.

[3] 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*. 2018, doi: 10.1002/cyto.a.23249.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015, doi: 10.1007/978-3-319-24574-4_28.

[5] OpenCV, "Contours : Getting Started." https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html.

[6] OpenCV, "Contours Hierarchy." https://docs.opencv.org/3.4/d9/d8b/tutorial_py_contours_hierarchy.html.

[7] "Chapter 9: Cells divide during the mitotic phase." https://bodell.mtchs.org/OnlineBio/BIOCD/text/chapter9/concept9.3.html.