Tracking of biological cells in time-lapse microscopy images

Simon Lin
Computer Science and Engineering
The University of New South Wales
Sydney, Australia
z5193123@unsw.edu.au

Abigail Sarmiento
Computer Science and Engineering
The University of New South Wales
Sydney, Australia
a.sarmiento@student.unsw.edu.au

Kovid Sharma
Computer Science and Engineering
The University of New South Wales
Sydney, Australia
z5240067@ad.unsw.edu.au

Andy Tran
Computer Science and Engineering
The University of New South Wales
Sydney, Australia
z5084520@unsw.edu.au

***Abstract*—We explore automatic segmentation, tracking, and mitosis detection for biological cells. Segmentation was approached with deep learning techniques that are robust and can be applied to datasets without preprocessing, tracking uses a mixed hybrid machine learning and naive approach, while mitosis involves mostly a naive approach.**

***Index Terms*—computer vision, biology, bioinformatics, cell, tracking, machine learning, classification, neural network**

I. Introduction

Cells undergo complex interactions through the simple mechanics of moving and dividing. These mechanics can be captured with imaging techniques [1] and have been analysed by human observers to be able to draw insights into biological processes. Throughout the past few decades, efforts have been made to run this analysis automatically, with early attempts relying on naive algorithms [2] that are derived from human observations as learned patterns or templates. These naive algorithms don't have a guarantee of generalizability, involving many fine tuned parameters that could impede progress rather than assist biologists and researchers. In this paper, we propose extending deep learning methods to generalize as much as possible for the tasks of cell segmentation, tracking, and mitosis detection. This paper is organised as follows: Section II presents a study on previous attempts at automated cell detection in the field including both naive and machine learning approaches. Section III gives an overview of the various methods that were tested with regards to each of the tasks. In Section IV, we describe our experimental set-up and approach, where the datasets, application of methods, and evaluation methodology are introduced. A discussion of the qualitative and quantitative results are provided in Section V, before concluding with a brief summary on the strengths, limitations and future research directions in Section VI.

II. Literature Review

Naive segmentation attempts based on patterns and human-learned observations were first attempted in [1]. A considerable amount of research has already been done on automatic cell detection in the past years. In most optical microscopy, images are captured by reflection and absorption of light. Many of the earlier methods were developed primarily for use with fluorescently labelled cells, which produce high-contrast images. Fluorescence microscopy is based on the phenomenon of fluorescence. Majority of cell-tracking systems available are designed primarily for use with fluorescently labelled images. We've also been given a phase-contrast dataset whose image sequences have a bright or dark ring(s) around the cells and also a differential interference contrast dataset. DIC works by separating a polarized light source into two orthogonally polarized mutually coherent parts. In previous methods, detection and tracking multiple spot-like objects was done for images captured by multidimensional fluorescence microscopy and PhC only. The method described in [1] works on PhC and Fluorescence microscopy but not on DIC. This makes this method limited to only two datasets. Another limitation of this method is mitosis detection in the post-processing step using a template matching-based tracking, which is applied in the backward direction in the spatio-temporal domain. Ergo, recovering trajectories is needed to connect the broken trajectories of cells in the spatio-temporal domain. A low accuracy score for tracking, overhead for trajectory recovery, high time cost and limitation to two datasets make this method less effective.

With the increasing power of computer technologies, deep learning methods, such as convolutional neural networks (CNN), are increasingly being adopted in image analysis. An interesting method for cell segmentation utilises two CNNs to predict markers and regions which are then transformed using watershed mathematical operations. This hybrid method was able to provide state-of-the-art performance on segmenting images with dense cell populations like our own. Hence, a deep learning approach are a potential area to explore. In regards to mitosis detection, Yang et al. have proposed a naive manner by employing the characteristic features of cells such as their area, perimeter, intensity and circularity between frames to identify candidate cells as they go through the various stages of mitosis. Likewise, Irshad et al. identify cell candidates as either being mitotic or non-mitotic using classifiers such as decision tree and Support Vector Machine (SVM) through extracting cell features such as colour and texture (e.g. Haralick) and utilising methods such as Scale Invariant Feature Transform (SIFT). Automated detection of mitosis event have also been achieved using deep neural networks with varying levels of results.

To explore ways to overcome these shortcomings or reap the benefits of performance, we will be applying naive and deep learning techniques to the tasks of segmentation, tracking and mitosis for all three datasets.

III. Methods

A. Segmentation

Deep learning techniques, as described below, were explored alongside naive algorithms to explore the problem of automatic segmentation of cell datasets.

1) Mask R-CNN: An approach to cell segmentation involved using Mask R-CNN [3] and transfer learning to the cell dataset from COCO [4] pretrained weights. R-CNN stands for Regions with Convolutional Neural Networks [5], which has the role of segmenting regions naively and leveraging CNNs to classify regions. Fast R-CNN [6] builds on top of R-CNN by reducing training down to a single stage by introducing the Region of Interest (RoI) pooling layer idea. Faster R-CNN [7] builds on top of Fast R-CNN by introducing the idea of "Region Proposal Networks" which is essentially generating the proposed RoIs with a neural network. Mask R-CNN is a state-of-the-art technique based on top of Faster R-CNN which adds an additional RoI training branch to allow for mask and instance segmentation. Mask R-CNN is available as a Tensorflow v1 library [8], with implementations in other machine learning frameworks [9].

2) U-Net: U-Net [10] is a fully convolutional network (FCN) which was developed for the purpose of segmenting biomedical images in mind, similar to that of this study and, hence, why it was considered for examination. Its network structure is similar to auto-encoders but with extra skip connections between layers to allow more features be passed to the up-sampling stage. The framework employs two paths. The first is used to classify the key features (i.e. the ”what”) of an image through a sequence of convolutions, which extract features through a number of filters, and max-pooling, to select those that are the most significant, operations in a contracting network. From this phase, a large number of high-resolution features are obtained, with each step doubling the amount of components in each feature vector. To localise (i.e. the ”where) these features, this path undergoes expansion. In this expansion path, up-sampling is employed, effectively mapping the feature vector of each pixel to a window in the output image, as well as further ”up-convolutions” which halve the quantity of features back down at each step [11]. The two paths are symmetric, with the former undergoing four stages of downsampling and the latter experiencing four stages of upsampling, and so the network forms a ’U’ shape of which it derives its name from. U-Net has been widely adapted to form other U-Net based architectures due to its outstanding performance from 3D U-Net [12] which uses 3D operations instead of 2D to a self-adapting version known as nnU-Net [13].

3) JNet: JNet [14] is a variation of U-Net for image segmentation with similar functionality. Comparing with U-net, JNet only contains a upsampling network with input image of different resolutions being directly feed into each convolution layer. Data-augumentaion is also used for training to solve the issue of lacking training images. We trained a JNet segmentaion network for DIC dataset only by using 16 ground truth images.

B. Tracking

1) DeepSort: An approach to cell tracking involved using DeepSort [15] [16]. DeepSort uses a Kalman filter [17], cosine similarity based on a Convolutional Neural Network, and the Hungarian Algorithm [18] (linear assignment) for ID association to the respective bounding boxes of cells. The implementation used is adapted from NanoNets [19].

2) OpenCV: Another approach to cell tracking involved using the centroid tracking algorithm [20] as our approach for Tracking analysis for all 3 datasets. A centroid tracker is a type of segmentation-driven method for tracking, it uses the distances between object centroids from two subsequent frames to determine the trajectory of each object. The centroid tracking algorithm can be described as follows:

Algorithm 1: Centroid Tracking

```
for each segmented object $S$ in frame $N$ and $N+1$ do
    compute the centroid for the object;
for each centroid $C$ in frame $N$ do
    pair object's centroid with another unpaired
      centroid from frame $N+1$ that has the shortest
      distance to $C$;
The distance between the 2 paired centroids represents
  the distance the object have travelled between the 2
  frames;
Object IDs will be assigned or removed based on the
  number of total objects detected from frame $N$ and
  $N+1$;
```

Although we are able to achieve good tracking results using centroid tracking for all datasets, the method also comes with a few limitations:

(i) It assumes every objects detected have the same movement speed.

(ii) The tracking is purely based on the distance between centroids. Other factors such as texture, deformation and intensity are not used for tracking.

(iii) The method will not do well on fast-moving objects where the centroids from the same object are far away from each other between each frame.

(iv) Because the segmented results for each frame are used for centroid tracking, the accuracy of the segmentation is crucial in obtaining a good accuracy for tracking.

C. Mitosis Detection

1) Naive Bounding Box: Naive non-deep learning approach is adopted for mitosis detection for each of the three datasets based on different rules and parameters. For each dataset, we are able to observe certain patterns from a mitosis cell that is visually different from normal cells. Hence, we adopted a rule-based method to detect mitosis events for all three datasets. In our implementation, we detect the enter point and exit point for a mitosis event of one single cell. This means for each segmented cell, we will pass it through our mitosis-detection rules to evaluate the cell's mitosis status, and this is conducted for every frame in the sequence.

Mitosis detection for DIC dataset: We have observed that a cell that is undergoing mitosis would be circular shaped with a higher average intensity. After the mitosis is completed, two daughter cells with smaller area will be created. So in order for a segmented cell to be classified as undergoing-mitosis it needs to satisfy that:

- i_o - i_f $>$ I and;
- Object area $>$ A and;
- Compactness $>$ C;

where i_o is the average intensity of the object, i_f is the average intensity of frame, and I,A,C are our defined threshold parameters which will be explained in the next paragraph.

We have experimented that applying a single fixed threshold I for object intensity filtering may not be suitable because of the inconsistent brightness across different sequences and frames, potentially leading to more false positives and false negatives. Hence intensity parameter I is apply on the difference between object intensity and frame intensity to reduce detection errors. For DIC, we set I = 2. As compactness measures the circularity of the object, a perfect circle would have compactness = 1. After testing with different values, we set C = 1.45. The area threshold A is used for filtering out smaller cells being incorrectly classified as mitosis. We set A = 7500 for DIC.

In addition, our algorithm also detects whether a mitosis is completed. This is simply achieved by calculating the ratio of area of the same mitosis-cell between 2 frames. If the ratio drops below certain threshold R, the mitosis is labelled as completed. We set R = 0.8 for DIC.

Mitosis detection for Fluo-Hela dataset: Hela cells captured in fluorescent light have an elliptical shape. Few frames before mitosis they grow into a round shape. They then elongate into a figure '8' shape out just before mitosis and after mitosis turn back into either elliptical or round shape with a smaller size. Therefore, our criteria for a mitosis event being detected is given as:

- Object area $>$ A and,
- Compactness $<$ C.

The defining characteristic for mitosis is that distinct change of shape. By adjusting a maximum compactness value of C = 1.17, we make sure that mitsos is verified if potential mitotic cell is as close to being a perfect ellipse. Similarly to DIC, we set a minimum area A = 221 to ensure that undesired small cells are not detected. To signify that a mitosis has finished another set of rules has been administered:

- Ratio of current area to previous area (at mitosis) $<$ A,
- Compactness $>$ C.
- Current intensity $<$ Previous intensity

Mitosis detection for PhC dataset: Based upon visual observation, it can be discerned that just before undergoing mitosis, a PhC cell quickly forms a near-circular shape before slowly elongating and splitting out into two smaller teardrop shaped cells, which are similar in form to its parent. Therefore, our criteria for a mitosis event being detected is given as:

- Object area $>$ A and,
- Compactness $<$ C.

The defining characteristic for mitosis is that distinct change of shape. By adjusting a maximum compactness value of C = 1.20, we ensure that a potential mitotic cell is as close to being a perfect circle, whilst also providing some leeway for outlier cells that do not necessarily take this exact form. Similarly to DIC, we set a minimum area, A = 50, to ensure that undesired small cells are not detected. On the other hand, whilst detection for DIC involves a parameter to compare the relative intensities between cell and frame, when a PhC cell is first observed to undergo a split, its average intensity is fairly stable. Hence, this criteria was not considered for this dataset. To signify that a cell has completed its division, another set of rules has been applied which are:

- Ratio of current area to previous area (at mitosis start) $<$ A, and
- Current intensity $<$ 1.1 x Previous intensity (at mitosis start).

The former was defined since cells that perform mitosis mutate into two smaller children cells in which a ratio of 0.85 (i.e. a split cell is assumed to be at most 85% of the size of its parents after the split) was prescribed. It can also be observed that cells adopt a wispy texture and faded intensity after mitosis and so this trait is enforced by checking whether the intensity of the cell at the current time has been reduced to below its a factor of 1.1 times its intensity at the start of division.

IV. Experimental Setup

The datasets used were gathered directly from the Cell Tracking Challenge website [21]. Specific datasets used for the results are as detailed:

DIC-C2DH-HeLa are “HeLa cells on a flat glass”, where HeLa are human-derived “immortal cells”. The cells exhibit mostly stable behaviour, only undergoing mitosis from time to time.

Fluo-N2DL-HeLa [22] is fluorescently labelled human-derived cells: the same type of cells as DIC-C2DH-HeLa, but captured with a different process and microscope. This dataset contains mostly washed out images that require contrast enhancements in order to view cells correctly.

PhC-C2DL-PSC [23] is “Pancreatic stem cells on a polystyrene substrate”. The background has a radius gradient from light to dark that could affect segmentation techniques. As the dataset progresses in sequence, the number of cells increase as mitosis occurs at a progressively higher rate.

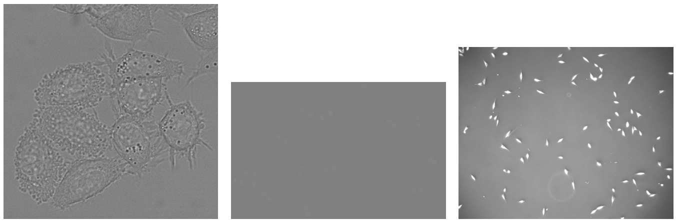
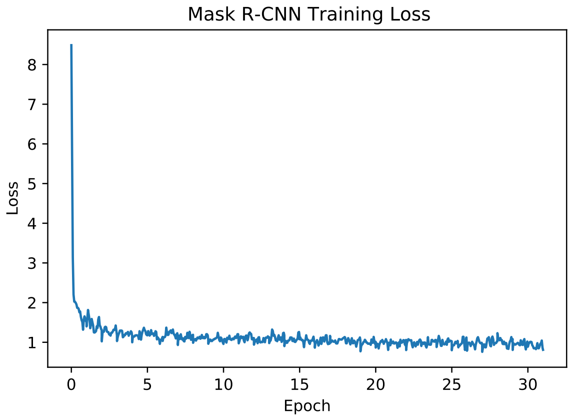


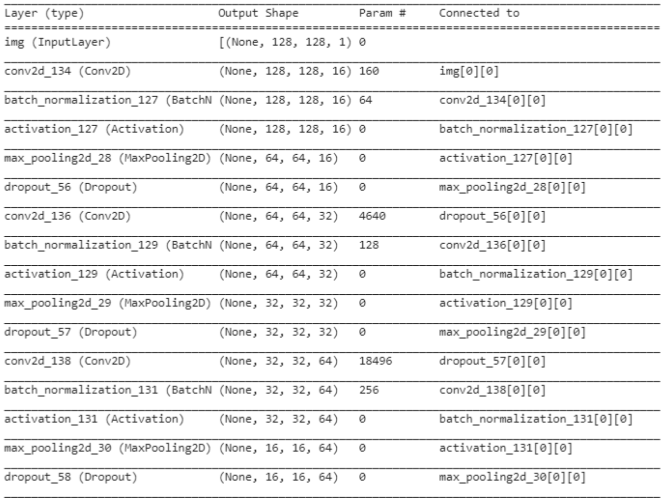
Fig. 1. Left to Right: DIC-C2DH-HeLa, Fluo-N2DL-HeLa, PhC-C2DL-PSC

A. Segmentation

1) Mask R-CNN: As a deep learning network, Mask R-CNN required a training environment and a base implementation, alongside parameters for training. Training the deep learning models was performed on Google Colab, which offers a range of GPUs on its runtime, ranging from NVIDIA K80s to T4s and P100s. These devices offer 24GB of GDDR5, 16GB of GDDR6, and 16GB of HBM2 memory respectively, with a minimum of 2,560 CUDA cores available. Two Mask R-CNN models were explored, the Matterport Mask R-CNN library, and an implementation on PyTorch TorchVision [24]. Augmentation [25] is an important part of training such a deep learning model, as it can lead to improved model performance [26] [27], with the augmentations applied to our method involving rotation, scaling, gaussian blurring, flipping, and sharpening. It was discovered that the Matterport Mask R-CNN library had trouble training with augmentations, and hence the PyTorch Mask R-CNN implementation was explored instead. The optimiser is Stochastic Gradient Descent, and the learning rate is set to a constant 0.005 after the first epoch. The model was trained for 30 epochs using 75% of the dataset with 25% held out for validation every 10 epochs.



2) U-Net: An adaptation of the original U-Net infrastructure [28] was examined for the purposes of segmentation, particularly with the PhC-C2DL-Hela dataset. A total of 101 images was used to train on with 101 corresponding masks which provided a ground truth values (silver truths were used as only two gold ground truths were provided) of the labelled segmented cells from which the model can learn from. A 90:10 split was made between training and validation sets, and four testing sequences of around 300 to 400 images each was evaluated. A summary of the model is given in Figure 2.



```
____________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
====================================================================================================
img (InputLayer)                [(None, 128, 128, 1) 0
____________________________________________________________________________________________________
conv2d_134 (Conv2D)             (None, 128, 128, 16) 160         img[0][0]
____________________________________________________________________________________________________
batch_normalization_127 (BatchN (None, 128, 128, 16) 64          conv2d_134[0][0]
____________________________________________________________________________________________________
activation_127 (Activation)     (None, 128, 128, 16) 0           batch_normalization_127[0][0]
____________________________________________________________________________________________________
max_pooling2d_28 (MaxPooling2D) (None, 64, 64, 16)   0           activation_127[0][0]
____________________________________________________________________________________________________
dropout_56 (Dropout)            (None, 64, 64, 16)   0           max_pooling2d_28[0][0]
____________________________________________________________________________________________________
conv2d_136 (Conv2D)             (None, 64, 64, 32)   4640        dropout_56[0][0]
____________________________________________________________________________________________________
batch_normalization_129 (BatchN (None, 64, 64, 32)   128         conv2d_136[0][0]
____________________________________________________________________________________________________
activation_129 (Activation)     (None, 64, 64, 32)   0           batch_normalization_129[0][0]
____________________________________________________________________________________________________
max_pooling2d_29 (MaxPooling2D) (None, 32, 32, 32)   0           activation_129[0][0]
____________________________________________________________________________________________________
dropout_57 (Dropout)            (None, 32, 32, 32)   0           max_pooling2d_29[0][0]
____________________________________________________________________________________________________
conv2d_138 (Conv2D)             (None, 32, 32, 64)   18496       dropout_57[0][0]
____________________________________________________________________________________________________
batch_normalization_131 (BatchN (None, 32, 32, 64)   256         conv2d_138[0][0]
____________________________________________________________________________________________________
activation_131 (Activation)     (None, 32, 32, 64)   0           batch_normalization_131[0][0]
____________________________________________________________________________________________________
max_pooling2d_30 (MaxPooling2D) (None, 16, 16, 64)   0           activation_131[0][0]
____________________________________________________________________________________________________
dropout_58 (Dropout)            (None, 16, 16, 64)   0           max_pooling2d_30[0][0]
____________________________________________________________________________________________________
```

Fig. 2. Summary of first few layers of the U-Net model

An initial quantity of 16 filters was administered and after 5 steps of convolutions a feature size of 256 is obtained. It can also be realised that the max pooling operations reduces the size of the image by a factor of 2 each time it is applied, which is then replaced by a series of transposed convolutions as the model undergoes its expansion phase. To monitor the performance of a model, a binary cross entropy loss was calculated at each epoch. A total of 75 epochs was fitted on the model with the epoch with the best validation log-loss of 0.1150 (see Figure 4) and accuracy of 0.6225 being adopted by the final model for evaluation and prediction. The graph indicates that the model steadily improves without overfitting the data.

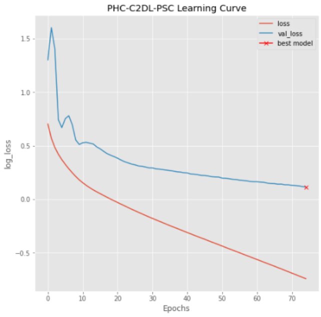


Fig. 3. Learning rate of the U-Net model on the Phc data

Similarly, it was used for Fluo-N2DL-Hela dataset. A total of 75 epochs was fitted on the model with the epoch with the best validation log-loss of 0.1753 (see Figure 4) and accuracy of 0.7616. The graph shows a linear log loss, this is due to the fact that the masks used for the testing weren't of good quality. On visual inspection, the segmented Silver Truths available for this dataset were not great and therefore rejected.

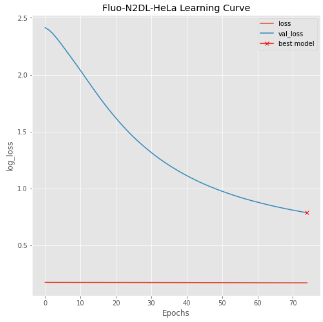


Fig. 4. Learning rate of the U-Net model on the Fluo-Hela dataset

3) JNet: A JNet segmentation network is trained for segmenting DIC images. The network is trained on Colab Pro using the labelled gold truth images for training. The training process took 14 hours using Tesla P100. Data augmentation techniques is used to provide more training data and increase the robustness of the network. Similar to U-Net, the network outputs a grey-scale image as the mask for each frame. Because JNet is a semantic segmentation network which does not explicitly label each segmented object, we use dice-coefficients and pixel-level difference-plots to evaluate segmentation results from JNet.

4) Naive Methods: Naive methods to perform cell separation was carried out on the Fluo-N2DL-HeLa and PhC-C2DL-PSC sequences due to their relatively simple nature (i.e. they both had no complex topology such as the existence of cells within cells).

For Fluo cells, we performed cell segmentation by using Adaptive Thresholding, with gaussian-weighted sum of the neighbourhood values minus the constant C. Otsu and Otsu-thresholding were first tried for this method but they did not lead to expected results. The image obtained with adaptive thresholding is then used to find the contours with the help of Canny Edge detection. Sometimes cells are at the boundary with only a part of them visible, this makes it harder to do segmentation and tracking and leads to false positives. To solve this, a border of 10 pixels around the frame is rejected.

As for the PhC cells, because of the presence of lighting artefacts and random noise, a top-hat filter, which applies a minimum then a maximum filtering, enabled the cell objects to be effectively separated from their background through a subtraction. The resultant image is contrast stretched between the whole range of intensities from 0 to 255 to achieve a greater dynamic range before thresholding is used which isolates the cells in the foreground into a binary image. Through experimentation, a threshold of between 75 and 255 was set, which ensured that most cells were obtained in the image without capturing the background lighting effects which are present in this dataset.

5) Evaluation Methodology: The Intersection over Union (IoU, otherwise known as the Jaccard Index for Bounding Boxes) was calculated between bounding boxes of the segmentation results and either the silver truths or gold truth bounding boxes from the Cell Tracking Challenge website [21].

$$IoU(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

IoU has a range between zero to one, with zero indicating no overlap and one indicating overlapping bounding boxes. In this instance, overlapped bounding boxes with a low IoU value (< 0.5) aren't discarded as false positives unlike required for the PASCAL VOC challenge [29]. Only IoU values of zero are regarded as there being no match, hence the corresponding ground truth box would be considered a false negative. Generalised IoU (GIoU) [30] is not suitable for this purpose as it avoids resulting in a value of zero, even when two compared bounding boxes have no overlap. As the association is made between segmentation bounding boxes and ground truth bounding boxes, those matches are marked as a ”true positive” result, whereby the segmentation bounding box is removed from future matching with other ground truth bounding boxes. At the end of the association, segmentation bounding boxes unmatched with a ground truth bounding box are labelled as ”false positives”. Similarly, ground truth bounding boxes not matched with a segmentation bounding box is labelled as a ”false negative”. This process allows for the automated analysis of segmentation accuracy using only ground truths and allows for the manual counting process to be skipped.

Dice-coefficient is another evaluation metric very similar to IoU for segmentation accuracy measurement. In our experiment, we use Dice Coefficient for evaluating JNet segmentation accuracy against the labelled silver-truth images from cell tracking challenge website. The formula for calculating dice score is:

$$Dice(A,B) = \frac{2 * TP}{FP + FN + 2 * TP}$$

where TP = number of true positive pixels that are correctly segmented; FP = number of false positive pixels from predicted mask; FN = number of false negative pixels from our predicted mask;

For each image sequence from DIC dataset, we calculated the average Dice Coefficient. The results is included in the results section.

To visually evaluate the accuracy of our JNet segmentaion on DIC dataset, we have plotted the difference-plot for our JNet results. A difference plot has the same shape as the original image with each pixel is color-coded by whether the pixel is classifed as TP, FP or FN. In our implementation, we use the following color-coding: grey = TP pixel; black = False Negative; white = False Positive;

B. Tracking

1) Evaluation Methodology: Evaluating tracking involved using the TRACK (TRA) gold truth dataset provided by the Cell Tracking Challenge website [21].

A custom script was written to calculate the MOTA [31] based on a comparison between the tracker assigned IDs to bounding boxes and the TRA dataset ground truths.

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t))}{\sum_t GT_t}$$

Where IDSW is the number of “identity switches” where the tracker assigns a different ID to a cell than what it previously assigned to the same ground truth.

C. Mitosis Detection

$$Precision = \frac{TP}{TP + FP}, \; Recall = \frac{TP}{TP + FN}$$

Mitosis detections were visually inspected manually and counted for true positives/false positives/true negatives/false negatives. Precision and Recall scores are calculated from these manual labelled data. This approach is taken due to there being no available ground truth available for mitosis detection. For PhC dataset, we are only able to manually annotate our mitosis results up to frame-50 due to the extensive labour required to annotate more crowded frames.

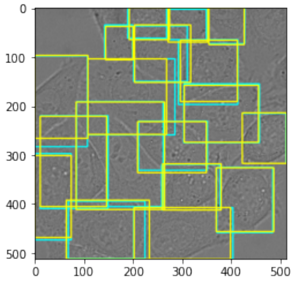
V. Results and Discussion

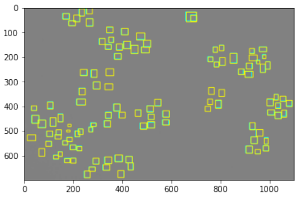
A. Segmentation

1) Mask R-CNN: Figure below: The boxes in blue are bounding box estimations from the segmentation algorithm (Mask R-CNN), while the yellow bounding boxes are derived from the ground truths. A yellow bounding box that are not associated with a blue bounding box is a false negative, while a blue bounding box not associated with a yellow bounding box is a false positive.

Left: Results for DIC-C2DH-HeLa/01/t083.tif against silver truth; false positives = 0, false negatives = 0, true positives = 18, average IoU for all true positives = 0.9564 (4dp).

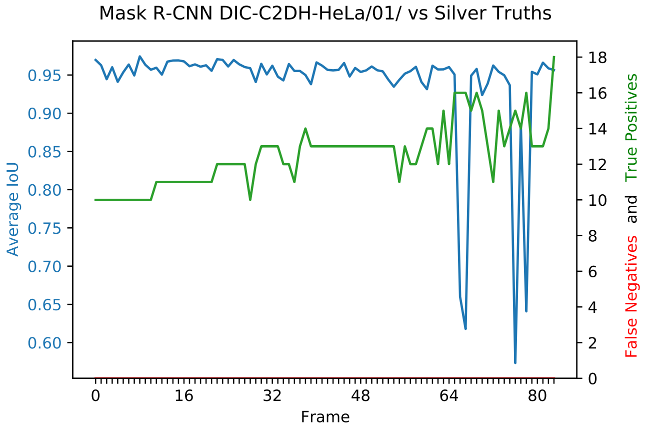
Right: Results for Fluo-N2DL-HeLa/01/t052.tif against gold truth; false positives = 0, false negatives = 5, true positives = 99, average IoU for all true positives = 0.7599 (4dp).



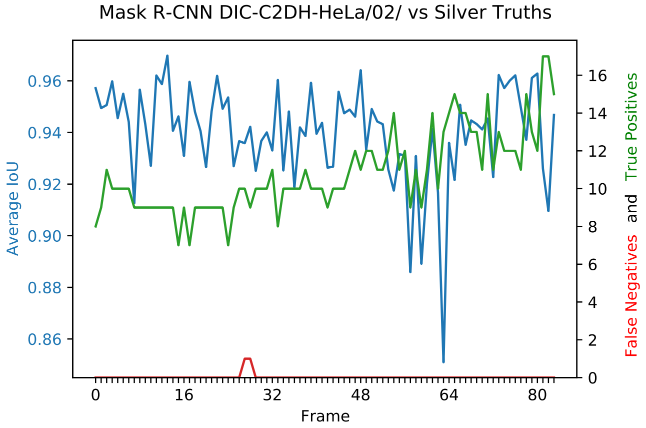


Validation accuracy uses the COCO evaluation metric [4].

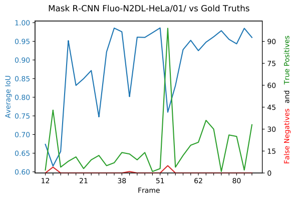
- Epoch 10:
  - Box: Average Precision (AP) @[IoU=0.50:0.95 — area= large — maxDets=100] = 0.760
  - Mask: Average Precision (AP) @[IoU=0.50:0.95 — area= large — maxDets=100] = 0.746
- Epoch 20:
  - Box: Average Precision (AP) @[IoU=0.50:0.95 — area= large — maxDets=100] = 0.763
  - Mask: Average Precision (AP) @[IoU=0.50:0.95 — area= large — maxDets=100] = 0.843
- Epoch 30:
  - Box: Average Precision (AP) @[IoU=0.50:0.95 — area= large — maxDets=100] = 0.791
  - Mask: Average Precision (AP) @[IoU=0.50:0.95 — area= large — maxDets=100] = 0.844

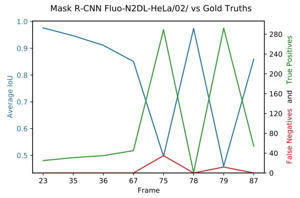

Mask R-CNN DIC-C2DH-HeLa/01/ vs Silver Truths

Mask R-CNN Segmentation on DIC-C2DH-HeLa/01/ resulted in low false negatives. Some average IoU variations as the frames progressed.


Mask R-CNN DIC-C2DH-HeLa/02/ vs Silver Truths

Mask R-CNN Segmentation on DIC-C2DH-HeLa/02/ resulted in similarly low false negatives as DIC-C2DH-HeLa/01/. With low average IoU variation throughout.


Mask R-CNN Fluo-N2DL-HeLa/01/ vs Gold Truths


Mask R-CNN Fluo-N2DL-HeLa/02/ vs Gold Truths

Mask R-CNN Segmentation on Fluo-N2DL-HeLa/01/ was compared against the dataset’s gold truth, which has abruptly labelled cells, hence the peak in true positives at frame 52. Apart from frame 52 and the beginning frames, the average IoU is fairly high. Mask R-CNN Segmentation on Fluo-N2DL-HeLa/02/ was sparse due to the low number of gold truths. There exists an inverse relation between the true positives and the IoU.

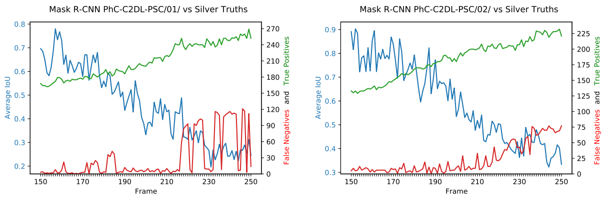


TABLE I
SEGMENTATION ACCURACY OF MASK-RCNN

	TP	FP	FN	Precision	Recall
DIC-C2DH-HeLa	1957	2	142	0.998	0.932
Fluo-N2DL-HeLa	1238	57	3124	0.956	0.282
PhC-C2DL-PSC	39730	5243	140	0.883	0.996

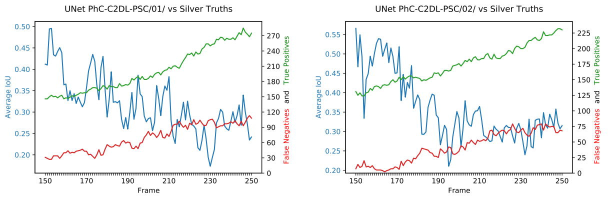
PhC-C2DL-PSC is where Mask R-CNN struggles, with "average" average IoU values initially for PhC-C2DL-PSC/01/ against its silver truths, but progressively decreasing in IoU scores. This may be due to the high growth in cells. PhC-C2DL-PSC/02/ tells a similar story to PhC-C2DL-PSC/01/. False negatives increase at a similar rate as the true positives.

The automated analysis does not reveal overfitting due to there being no ground truth associations for the test datasets on the Cell Tracking Challenge website [21], however there is evidence of minor overfitting as the visual segmentation results are slightly poorer on the testing datasets.

Overall, Mask R-CNN does a good job in segmenting the dataset with high average IoU values for the DIC-C2DH-HeLa and Fluo-N2DL-HeLa datasets. This result is noteworthy particularly as no preprocessing nor parameter tuning (across the different datasets) is involved for Mask R-CNN cell segmentation.

2) U-Net: On a qualitative basis, the predictions created by the model seems to be well fit. Individual cells are able to be distinguished from each other with a corresponding shape and in the correct position. Also, despite the PhC-C2DL-PSC data suffering from inconsistent lighting, the cell towards the middle of the image where the lighting is at its highest intensity are still detected to a workable quality by ensuring that an appropriate minimum threshold (0.35 in our case) is applied. This is all without the need for image processing, which enables a more optimal performance rate.

However, quality appears to drop out at the later frames wherein a greater density of cells populate the image. Because of the increased number of objects and their close proximity to one another, some cell shapes merge with another cells forming compound contours which makes it more problematic to isolate the separate cells, thus, lending to an erroneous segmentation. Additionally, on account of their diminishing structure and intensity (i.e. the 'newer' cells created from cells which have undergone mitosis tend to be of more airy in appearance), some cell shapes sometimes go undetected and are lost in the resulting mask. A similar trend can be realised when looking at the IoU values for the PhC-C2DL-PSC dataset in which average results were obtained for both image sequences, compared to that of Mask R-CNN, with a maximum IoU of approximately 0.55. This value continues to fall through the frames as the build-up of cells grows exponentially rendering it a more difficult task to accurately segment the image, as was previously described. Other reasons for the falling performance, despite the popularity of this deep learning model for biomedical image segmentation, may be attributed to the fact that the model was only trained with a limited quantity (i.e. 100) of labelled images whereas other studies [32] may use up to the order of thousands of projections. Thus, it is suggested that where target information is scarce, data augmentation techniques should be utilised to provide a more robust dataset for training.



3) JNet: We have computed the average Dice Coefficient for JNet using labelled silver-truth available for sequence 1 and sequence 2 of DIC dataset, compared with our Mask-RCNN network and another pretrained TUG-AT [33] network:

TABLE II
DICE COEFFICIENT OF JNET, MASK-RCNN AND TUG-AT BASELINE ON DIC

	JNet	Mask-RCNN	TUG-AT
Sequence 1	0.9291	0.9477	0.9287
Sequence 2	0.9199	0.9250	0.9171

It is surprising the dice score from both Mask-RCNN and JNet is very close to the baseline TUG-AT network. Especially for JNet as it was only trained with 20 labelled images. To further visually illustrate the accuracy of JNet segmentaion, the 2 images below shows a example of segmented result from JNet with its difference-plot against ground truth:

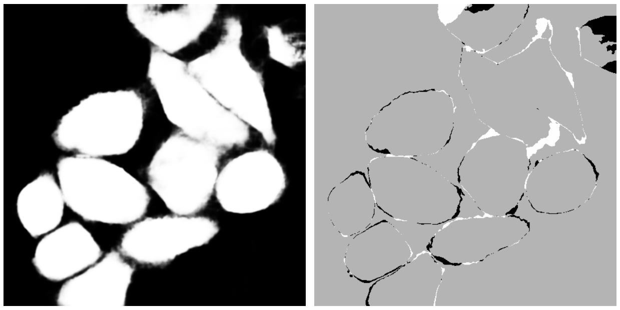
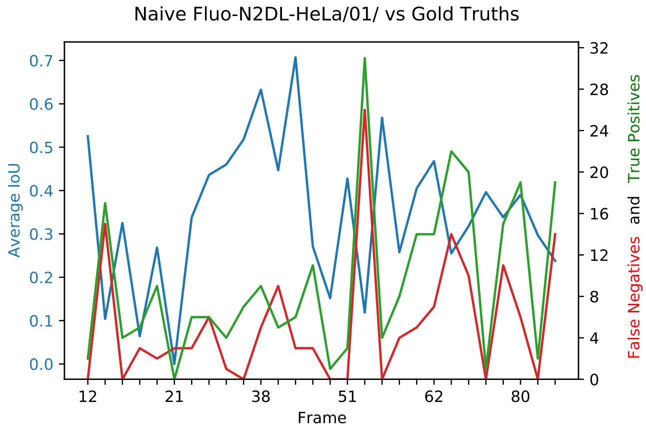


Fig. 5. JNet result and its difference plot against ground truth

The JNet result is very close to the actual ground truth with the majority of pixels are correctly classified. Although some false positives and false negatives are detected around the edge of each object, such difference can be reduced by post-processing the masks, which we did not have enough time to implement.

Remarks on JNet/U-Net: Different from Mask R-CNN which is designed for instance segmentation, a U-Net-like structure can only be used for semantic segmentation which is not able to identify each object. For DIC dataset in particular, because cells are clustered together, the mask output from these networks has to be post-processed before being used for tracking and mitosis despite good accuracy is achieved. For the same reason, a U-Net-like network might be more suitable for tasks to segment object from a complex image as in [34].

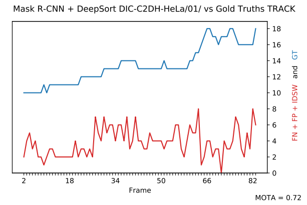
4) Naive Methods: As a baseline, a naive segmentation approach was attempted with Fluo-N2DL-HeLa.

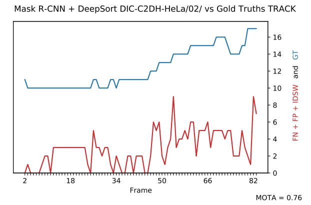


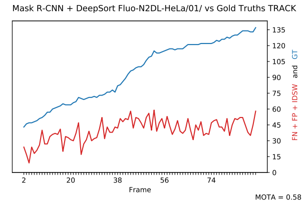
This resulted in a higher false negative, lower true positive, and lower average IoU scores than the Mask R-CNN segmentation attempt, this implies that preprocessing results in lost cell information. The segmentation techniques applied for the PhC dataset proves to perform comparitively well even to the neural network approach such as U-Net. Because the images aren't overly complex, simple morphological operations and thresholding allow the image to be segmented to an acceptable quality.

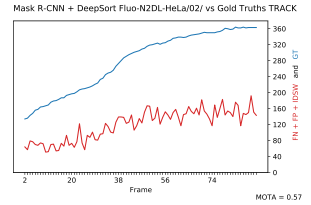
B. Tracking

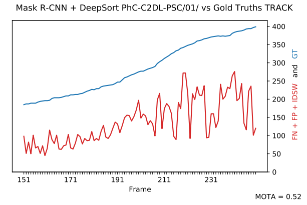
Running DeepSort on the Mask R-CNN segmentation bounding boxes resulted in similar MOTA values across all datasets.

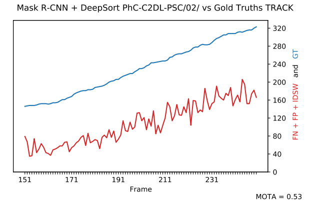




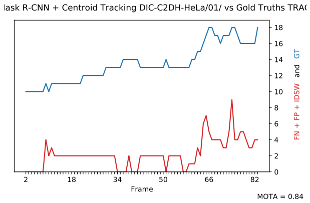


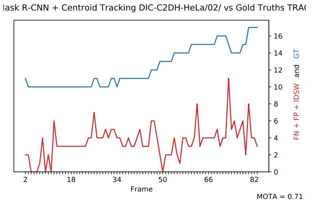






Running centroid tracking on the Mask R-CNN segmentation bounding boxes resulted in higher MOTA values than DeepSort for DIC-C2DH-HeLa/01/, however a lower (but similar) MOTA for DIC-C2DH-HeLa/02/.





The MOTA values are demonstrated to reveal fairly good tracking as MOTA is known to run into negative value ranges, which our tracking techniques do not encounter.

C. Mitosis Detection

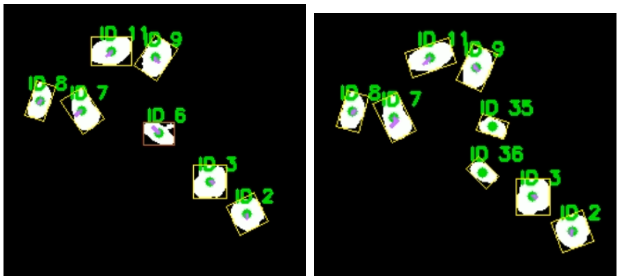


Fig. 6. Illustration of tracking and mitosis detection. (a) Tracking of cells in frame 3 of Fluo-N2DL-HeLa sequence, where cell 6 is undergoing mitosis. (b) Tracking of cells in frame 4 where mitosis has completed and new IDs assigned to daughters.

TABLE III
AVERAGE MITOSIS PRECISION AND RECALL SCORES FOR ALL DATASETS

	TP	FP	FN	Precision	Recall
DIC-C2DH-HeLa	21	11	6	0.778	0.656
Fluo-N2DL-HeLa	160	41	55	0.744	0.796
PhC-C2DL-PSC	44	16	33	0.571	0.733

We are able to achieve better mitosis-detection accuracy for DIC and Fluo than PhC. We argue that the low accuracy is because of the smaller size of PhC cell and also the intensity does not vary during mitosis. As we have only implemented mitosis detection by evaluating object features, more advanced approaches such as using image filtering trajectory recovery technique should be able to achieve better results.

VI. CONCLUSION

Different approaches were taken the tasks of segmentation, tracking, and mitosis detection, with the better performing results across all datasets relying on a combination of Mask R-CNN and centroid tracking. For segmentation of PhC-C2DL-PSC, U-Net results in metrics similar to Mask R-CNN. For tracking, DeepSort worked well, however required additional training to work better and a better mitosis detection algorithm paired with it. Our centroid tracking implementation had far better mitosis detection and track splitting features than our DeepSort experiments. In future, given more time and knowledge (and group members contributing), the Mask R-CNN results can be improved with model training techniques such as k-Fold and other overfitting prevention strategies. DeepSort can also be better explored to approach both segmentation and tracking with deep learning. The mitosis detections could also have been evaluated automatically with the TRA datasets from the Cell Tracking Challenge website and even potentially used to train a deep neural network to detect mitosis. In summary, deep learning has been incredibly useful for approaching this project, however we found that it is far from a user-friendly approach to biological dataset analysis.

VII. CONTRIBUTION OF GROUP MEMBERS

A. Simon Lin

Shared role in Mask R-CNN model training. Exploration, model training, and report writing on JNet including Dice coefficient analysis. Primary contributor to centroid tracking. Lead developer of the underlying centroid tracking and mitosis detection code that the team has chosen as the primary approach. Creation of a desktop interface to visualize results and to show the results of Task 3 including confinement ratio, distance cell travelled. Counting and evaluation of mitosis detection results.

B. Abigail Sarmiento

Initial responsibility, explorations, trials, and building blocks for research and development on U-Net that allowed the team to explore and utilize this approach. Primary contributor to U-Net report section. U-Net code testing, adaptation, and training for PhC-C2DL-PSC segmentation. Contributed to the analysis of PhC-C2DL-PSC for both naive and U-Net segmentation results. Centroid tracking adaptation for PhC-C2DL-PSC. Contributed to mitosis detection using centroid tracking.

C. Kovid Sharma

Trial of OTSU for cell detection. Testing and training dataset on U-Net. Implementing Adaptive thresholding for cell segmentation, addition of boundary cell rejection, centroid tracking adaptation for Fluo-N2DL-HeLa. Did manual analysis of Fluo-N2DL-Hela naive for segmentation and mitosis. Primary contributor to the literature review. Contributed to mitosis detection using centroid tracking.

D. Andy Tran

Solely performed all of the necessary adaptions and exploration Mask R-CNN (for both the Matterport and PyTorch versions) to allow training and inference to be possible for Task 1.1 across all datasets. this includes writing code for the necessary augmentation additions and dataset loading. Shared role in Mask R-CNN model training. Primary contributor to the Mask R-CNN related report sections. Wrote code and created tooling for automated IoU analysis based on ground truths and bounding boxes for segmentation results. Created tooling for the automated MOTA analysis for tracking results. Exploration of DeepSort for tracking and (partially) mitosis detection.

REFERENCES

[1] E. Meijering, O. Dzyubachyk, and I. Smal, "Methods for cell and particle tracking," in *Imaging and Spectroscopic Analysis of Living Cells - Optical and Spectroscopic Techniques*. Elsevier, 2012, pp. 183–200. [Online]. Available: https://doi.org/10.1016/b978-0-12-391857-4.00009-4

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

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2017.

[4] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[6] R. Girshick, "Fast r-cnn," 2015.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2015.

[8] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," https://github.com/matterport/Mask_RCNN, 2017.

[9] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[11] [Online]. Available: https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

[12] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2016, pp. 424–432.

[13] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert *et al.*, "nnu-net: Self-adapting framework for u-net-based medical image segmentation," *arXiv preprint arXiv:1809.10486*, 2018.

[14] "J-net: Multiresolution neural network for semantic segmentation," https://github.com/tsixta/jnet.

[15] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[16] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 748–756.

[17] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960. [Online]. Available: https://doi.org/10.1115/1.3662552

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

[19] Abhyantrika, "nanonets_object_tracking." [Online]. Available: https://github.com/abhyantrika/nanonets_object_tracking

[20] A. Rosebrock, "Simple object tracking with opencv," https://www.pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/, Jul. 2018.

[21] V. Ulman, M. Maška, K. E. G. Magnusson, O. Ronneberger, C. Haubold, N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic, I. Smal, K. Rohr, J. Jaldén, H. M. Blau, O. Dzyubachyk, B. Lelieveldt, P. Xiao, Y. Li, S.-Y. Cho, A. C. Dufour, J.-C. Olivo-Marin, C. C. Reyes-Aldasoro, J. A. Solis-Lemus, R. Bensch, T. Brox, J. Stegmaier, R. Mikut, S. Wolf, F. A. Hamprecht, T. Esteves, P. Quelhas, Ömer Demirel, L. Malmström, F. Jug, P. Tomancak, E. Meijering, A. Muñoz-Barrutia, M. Kozubek, and C. O. de Solorzano, "An objective comparison of cell-tracking algorithms," *Nature Methods*, vol. 14, no. 12, pp. 1141–1152, Oct. 2017. [Online]. Available: https://doi.org/10.1038/nmeth.4473

[22] B. Neumann, T. Walter, J.-K. Hériché, J. Bulkescher, H. Erfle, C. Conrad, P. Rogers, I. Poser, M. Held, U. Liebel, C. Cetin, F. Sieckmann, G. Pau, R. Kabbe, A. Wünsche, V. Satagopam, M. H. A. Schmitz, C. Chapuis, D. W. Gerlich, R. Schneider, R. Eils, W. Huber, J.-M. Peters, A. A. Hyman, R. Durbin, R. Pepperkok, and J. Ellenberg, "Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes," *Nature*, vol. 464, no. 7289, pp. 721–727, Apr. 2010. [Online]. Available: https://doi.org/10.1038/nature08869

[23] D. H. Rapoport, T. Becker, A. M. Mamlouk, S. Schicktanz, and C. Kruse, "A novel validation algorithm allows for automated cell tracking and the extraction of biologically meaningful parameters," *PLoS ONE*, vol. 6, no. 11, p. e27315, Nov. 2011. [Online]. Available: https://doi.org/10.1371/journal.pone.0027315

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8026–8037. [Online]. Available: http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[25] P. Simard, D. Steinkraus, and J. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* IEEE Comput. Soc. [Online]. Available: https://doi.org/10.1109/icdar.2003.1227801

[26] J. C. Caicedo, J. Roth, A. Goodman, T. Becker, K. W. Karhohs, M. Broisin, C. Molnar, C. McQuin, S. Singh, F. J. Theis, and A. E. Carpenter, "Evaluation of deep learning strategies for nucleus segmentation in fluorescence images," *Cytometry Part A*, vol. 95, no. 9, pp. 952–965, Jul. 2019. [Online]. Available: https://doi.org/10.1002/cyto.a.23863

[27] T. Falk, D. Mai, R. Bensch, Özgün Çiçek, A. Abdulkadir, Y. Marrakchi, A. Böhm, J. Deubner, Z. Jäckel, K. Seiwald, A. Dovzhenko, O. Tietz, C. D. Bosco, S. Walsh, D. Saltukoglu, T. L. Tay, M. Prinz, K. Palme, M. Simons, I. Diester, T. Brox, and O. Ronneberger, "U-net: deep learning for cell counting, detection, and morphometry," *Nature Methods*, vol. 16, no. 1, pp. 67–70, Dec. 2018. [Online]. Available: https://doi.org/10.1038/s41592-018-0261-2

[28] H. Lamba, "Understanding semantic segmentation with unet," Feb 2019. [Online]. Available: https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47

[29] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Sep. 2009. [Online]. Available: https://doi.org/10.1007/s11263-009-0275-4

[30] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2019. [Online]. Available: https://doi.org/10.1109/cvpr.2019.00075

[31] S. Anjum and D. Gurari, "CTMC: Cell tracking with mitosis detection dataset challenge," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, Jun. 2020. [Online]. Available: https://doi.org/10.1109/cvprw50498.2020.00499

[32] G. Landry, D. Hansen, F. Kamp, M. Li, B. Hoyle, J. Weller, K. Parodi, C. Belka, and C. Kurz, "Comparing unet training with three different datasets to correct cbct images for prostate radiotherapy dose calculations," *Physics in Medicine & Biology*, vol. 64, no. 3, p. 035011, 2019.

[33] M. F. H. B. M. U. Christian Payer, Darko Štern, "Segmenting and tracking cell instances with cosine embeddings and recurrent hourglass networks," pp. 106–119, 10 2019.

[34] P. Christ, F. Ettlinger, F. Grün, M. Elshaera, J. Lipkova, S. Schlecht, F. Ahmaddy, S. Tatavarty, M. Bickel, P. Bilic, M. Rempfler, F. Hofmann, M. D'Anastasi, S.-A. Ahmadi, G. Kaissis, J. Holch, W. Sommer, R. Braren, V. Heinemann, and B. Menze, "Automatic liver and tumor segmentation of ct and mri volumes using cascaded fully convolutional neural networks," 02 2017.