COMP700 Draft Proposal
Deep Learning for Tracking Moving Cells in Time-lapse Video Sequences

Alexander Goudemond, Student Number: 219030365

May 2022

Abstract

Cell segmentation and cell tracking is important in fields involving microscopy images. The ability to detect cell culture characteristics is valuable for those studying cellular or sub-cellular features, as well as those doing research around disease treatment, drug development, and cell changes and/or interactions over time. Cell segmentation is a challenging task, as each data-set can have distinct features and complications. Recent works have demonstrated that deep learning techniques for cell segmentation and cell tracking show promising results, and may improve upon traditional cell tracking techniques. This project aims to investigate and compare different cell segmentation techniques, as well as cell tracking techniques in 2D video sequences.

1 Introduction and background

Cell segmentation and cell tracking is valuable for several reasons. Examples include: disease research, tracking tissue growth, tracking cell changes over time (mitosis, apoptosis, etc.), cell velocity changes, lineage tracing, distinguishing between cellular and sub-cellular features as well as development towards drugs and vaccines. Software can greatly assist this work being conducted, and help automate repetitive tasks, which are normally done manually.

There are 2 types of microscopy images that can be used in cell segmentation and cell tracking programs: *fluorescence microscopy* and *transmission microscopy*.

Fluorescent microscopy encapsulates techniques around **fluorescent light**. A staining compound or gene (referred to as a 'tag' or a 'reporter') may be introduced into the cell. George [9] explains that high intensity light is then shone on the cells, and these tags reflect a lower energy light, with longer wavelengths. This property results in bright cellular features, against a dark background. Some cells/organisms may be naturally fluorescent, and a tag is not needed. However, many cells may require a tag.

Transmission microscopy encapsulates techniques around Transmission Electron Microscope **(TEM) images**. CCBER [2] demonstrates how a beam of electrons are shone onto specimens, resulting in a highly magnified and clear image . This microscope can create images that are Phase Contrast (PhC) (revealing refractive contrast over a specimen) or Differential Interference Contours (DIC) (introduces contrast to otherwise faint images)

Nuclear fluorescence microscopy is a reliable way to stain the nucleus of a cell. This is desirable as nuclei, across species, share a common shape and texture. This provides a good starting point for algorithm design, as most data-sets will contain easily visible nuclei. Thus, many algorithms explore nuclei detection and use these 'seed points' to identify the whole cell boundary. Some algorithms are only interested in nucleus detection, but whole cell detection is first achieved by nucleus segmentation.

Particle tracking is another application of cell segmentation and tracking techniques, however particles are easier to track. Most cells display *Brownian Motion*, making their movements in a time-lapse sequence erratic and unpredictable. Particles, on the other hand, rarely move and rarely interact with cells. In addition to this, particles often contain smooth boundaries, whereas many cells contain irregular boundaries, that grow and shift over time. Due to these differences, particle tracking is seen as a simpler task than cell tracking, and will not be the focus of this project.

2D cell tracking is much broader, and most of the research conducted already is on 2D data-sets. For example, Magnusson [7] and Wang, A et. al. [14] demonstrated the ability to track cells in 3D data-sets using dynamic iterative programming, and a Convolutional Neural Network (CNN), respectively.

There is also a difference between *global segmentation*, *semantic segmentation* and *instance segmentation*.

Semantic segmentation incorporates the **classification** of an item. It is useful in data-sets that contain many clearly distinct items, such as the difference between a car, a person and a cloud.

Instance segmentation is interested in **identifying individuals** in a data-set. Each boundary of an instance segmented image contains (at least) 1 individual inside of it, and the kind of individual it is, is not important to the model.

Global segmentation seeks to separate any objects with contrasting texture - irrespective of foreground or background.

Thus, the cell segmentation and cell tracking algorithms explored in this project focus on instance segmentation techniques.

Wang, J et. al. [15] mentions how cell tracking either takes the form of tracking by model evolution, or the form of tracking by detection. In the former, segmentation and tracking are addressed simultaneously, each frame. This is done by using some kind of hidden data, stored in a feature space. In the latter, the task is often split into cell detection and cell association. The tracking can then be achieved by comparing the 2 objects, and performing some kind of linking algorithm.

It is expected that the data-set provided to the program will contain cells that could be identified by a biologist. This is certainly the case with microscopy time-lapse data-sets, as the specimens are grown by the biologists, and thus the kind of cell must be known. However, artificial data-sets are being created, and in these situations the kind of cell is expected to be provided along with the data-set.

The Cell Tracking Challenge (CTC) [3], is an initiative to provide data-sets and algorithm metrics to the public. There is a need for open source software, and freely available data-sets - which the CTCs have the potential to provide. Data-sets from this site will be used in this project.

At the time of writing, CTC contains 10 publicly available 2D data-sets. These data-sets contain different kinds of cells, and further data-sets may be used in future CTC competitions. The 10 data-sets are shown in Table 1.
8 of these data-sets contain silver reference annotations, and 1 of the data-sets is simulated

This research extends existing work, and attempts to identify successful algorithms for broad level cell segmentation and tracking. The creation of software will abstract the algorithm, and enable researchers to focus on the results, without needing to understand or tweak the model. This research aims to create models that will optimise parameters automatically, resulting in a more user friendly program.

Figures 1 and 2 contain examples of cell segmentation and cell tracking, sampled from Magnusson [7].

Data-Set Description	**Data-Set Name**	**Cell Classification**	**Notes**
Mouse hematopoietic stem cells in hydrogel microwells	BF-C2DL-HSC	Mouse Stem Cells	
Mouse muscle stem cells in hydrogel microwells	BF-C2DL-MuSC	Mouse Stem Cells	
HeLa cells on a flat glass	DIC-C2DH-HeLa	HeLa Cells	DIC microscopy; tightly packed; complex intensity patterns; boundaries difficult to find
Human hepatocarcinoma-derived cells expressing the fusion protein YFP-TIA-1	Fluo-C2DL-Huh7	Derived Liver Cells	
Rat mesenchymal stem cells on a flat polyacrylamide substrate	Fluo-C2DL-MSC	Rat Derived Bone-Marrow Stem Cells	Fluorescence microscopy; irregular shape and size
GFP-GOWT1 mouse stem cells	Fluo-N2DH-GOWT1	Mouse Embryonic Stem Cells	Fluorescence microscopy; mixture bright and dim objects
HeLa cells stably expressing H2b-GFP	Fluo-N2DL-HeLa	HeLa Cells	Nucleur fluorescence microscopy; size and shape variability
Glioblastoma-astrocytoma U373 cells on a polyacrylamide substrate	PhC-C2DH-U373	Brain Tumour Cells	
Pancreatic stem cells on a polystyrene substrate	PhC-C2DL-PSC	Pancreatic Stem Cells	PhC Microscopy; sparse cells; complex intensity patterns
Simulated nuclei of HL60 cells stained with Hoescht	Fluo-N2DH-SIM+	(Simulated) Derived Blood Cells	Nuclear fluorescence microscopy; many cells; bright and dim regions

Table 1: Table showing the data-sets available on CTC website [3]

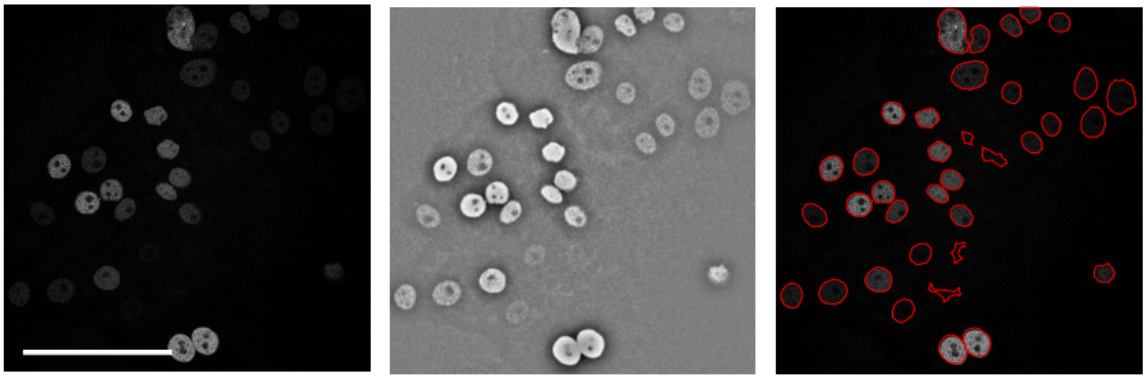


Figure 1: Example of Cell Segmentation - Sampled from Magnusson [7], pg17

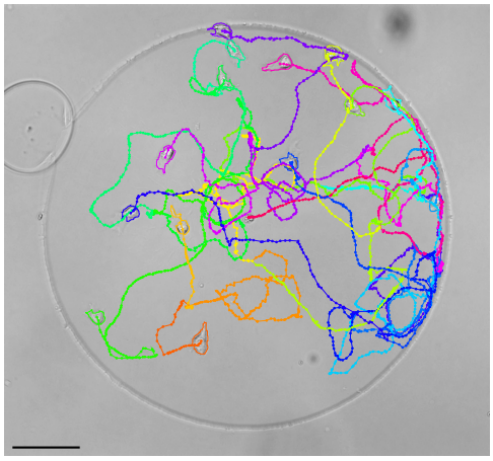


Figure 2: Example of Cell Tracking - Sampled from Magnusson [7], pg52

2 Research Problem

2.1 Motivation

Data-sets of cells are not equivalent. Each data-set contains different kind of cells, each with underlying properties and behaviours. Some data-sets contain stem cells, which tend to cluster as they grow, others contain cells that move frequently, and interact with their surroundings - resulting in an irregular and changing cell boundary. There are also data-sets that contain capturing issues - light saturation, camera shake, video-sequence time-delays, magnification adjustments and cell movement out of frame.

There is a need to highlight some CNN models already used, and show their pros and cons, trying to show the existing research gaps.

2.2 Problem Statement

There is a need for an exploration of how different deep learning algorithms track cells across data-sets. Existing research often uses high powered, and often unattainable, computers - so there is also a need to conduct this comparison on a more accessible computer.

The primary question that this research aims to answer is: "Which deep learning algorithm performs the best cell tracking, from a 2D time-lapse video sequence".

2.3 Aims and Objectives

The primary focus of this project will be implementing the algorithms across 10 data-sets, and then using the CTC metrics to rank and compare them. To achieve this, some broad level objectives are listed:

1. To review the state-of-the-art literature on the tracking of moving objects in time-lapse video sequences based on deep learning-based approaches
2. To segment moving cells using traditional techniques (Thresholding, Energy Minimization, etc.) as well as deep learning techniques (Nuclei Seed Detection, Distance Neighbours, etc.)
3. To model a deep learning framework (focusing on CNNs) for accurate cell tracking

3 Preliminary Literature Review

There are several benefits to a program being able to conduct cell segmentation and cell tracking. Some broad level benefits include automatic analysis of data-sets (increasing workflow and eliminating bias), the potential for cell feature tracking/detection in real time and even having a program that can interpret the results for the researcher.

Al-Kofahi et. al. [4] mentions how cell culture characterization is important with regards to cancer research and drug research. Cancer research benefits in particular because certain changes in cell features (like abnormal cell boundary changes) may be indicative of cancerous growth, mentioned also in Wang, A et. al. [14]. Magnusson [7] additionally mentions how some researchers need a program that can conduct lineage tracing, velocity tracking and tissue development.

Scherr et. al. [11], Magnusson [7] discuss how their software can also incorporate 3D data-set processing.

There are many tracking challenges that are faced with this kind of research. Magnusson [7] explains how software is not always designed to be universal; some aim to address a specific problem only. And some papers only focus on algorithm design, without ever implementing the software for use. This is compounded by the fact that many data-sets and ground truths are not publicly available. Scherr et. al. [11] and Magnusson [7] also mention a high computational need, with 64 GB of RAM needed to quickly process information. These resources are not always available to researchers, and algorithm comparisons using more accessible equipment is needed. Moen et. al. [8] also mentions how most applications of deep learning require supervised learning, with specialized training sets. This illustrates the difficulty with using unseen, live cell video sequences.

Other tracking challenges include cell clustering, joint segmentation, sample diversity and parameter tweaking, as outlined in Magnusson [7]. Fragmented cells, large Signal-To-Noise (SNR) images, annotation/label errors and unpredictable movement are mentioned in Scherr et. al. [11]. Wang, A et. al. [14] also mentions how some images require special pre-processing, like bright-field images, DIC images and PhC images.

Attempting to build collaborative software, Fazeli et. al. [6], discusses their attempt to build on top of/integrate with existing open-source software. Their program StarDist interacts with existing

software TrackMate, which is freely available to the public. They also mention how, although pre-trained models exist, they are likely to under perform on distinct data-sets. To address this, they offer the user a way to annotate their data-sets, through existing software Fiji. Although this is time consuming - it produces better results.

There are also biological challenges mentioned. For example, Magnusson [7] explains how staining in fluorescent microscopy images may result in drug reactions at cellular level. This may alter the cell's behaviour, or impact it's interaction with the environment. The staining may also be lost as a result of cell division, so it may not be ideal for lineage tracing.

Scherr et. al. [11] goes on to mention how apoptosis and movement out of frame of the video-sequence impacts the algorithms. These challenges are experienced at data-set creation, and need to be considered when the algorithms are designed. Additional pre-processing may be necessary, to handle things like: camera shake, magnification changes, time-sequence capture delays as well as dirty equipment - leading to smudges/blurred aspects in the data-set.

In terms of segmentation techniques, Al-Kofahi et. al. [4], Magnusson [7], Scherr et. al. [11], Moen et. al. [8] and Wang, A et. al. [14] mention how staining the nucleus allows for development of the initial segmentation steps. They explain how seed locations can be used as the approximate center point of each nucleus, and the whole cell boundary can be obtained as the next boundary. Different segmentation techniques are successful in this pursuit, though a popular one includes overlaying a mask onto the image, before applying a watershed segmentation technique.

Al-Kofahi et. al. [4] mentions how their segmentation includes single channel whole cell segmentation through deep learning, combining techniques from thresholding, the watershed algorithm and a kind of 'blob-detector'. This model needs to be trained offline before use. Other methods considered include active contour models, level-set methods, morphology based methods and snake algorithms. Magnusson [7] discusses how their dynamic iterative algorithm has different segmentation techniques based on whether it is a fluorescent or transmission image. It also incorporates ridge detection for data-sets that are clustered. Scherr et. al. [11] has a technique involving distance maps and the TWANG algorithm. Wang, A et. al. [14] mentions region growing as a segmentation technique. Ulman et. al. [13] mentions broad-level segmentation methodologies: thresholding, region growing, energy minimization, shape matching, edge detection and machine learning.

With regards to tracking techniques, Magnusson [7] employs a dynamic cell track-linking algorithm, which connects seeds between frames. This kind of detection algorithm is done using a nearest-neighbour approach, in terms of the smallest distance from the change in seed position.

Al-Kofahi et. al. [4] and Scherr et. al. [11] use a deep learning technique to track the movement of cells in a similar way.

Moen et. al. [8] implements a deep learning method using linear programming and a viterbi algorithm. It is also able to solve the cell-out-of-frame problem using 'shadow objects', which track the information about a cell's disappearance in each frame, and then tries to relate the information across frames.

Sun et. al. [12] provides an overview of the basic knowledge of deep learning and its applications. In particular, it mentions the popular deep learning techniques used for single-cell optical image studies.

In terms of Deep Learning models they draw attention to how CNNs and the Generative Adversarial Networks (GANs) have produced good results in image processing tasks. Additionally, popular Deep Learning methodologies for optical images include Transfer Learning, Multimodal Learning,

Multitask Learning and End-To-End Learning.

A valuable insight from Sun et. al. mentions the popular Fully Convolutional Networks: U-Net, SegNet and DeepLab. These architectures have been modelled after, or improved upon by many papers already.

The CTC competition, Ulman et. al. [13], provides detailed examples of factors that influence this research.

Looking at the quality of cell images and videos, one such comparison is the experience between SNR and Computed Radiography (CR). It is found that the best results occur when there is a high SNR and a high CR, with the worst results being a low CR and low SNR (The low SNR used was a seed density of 200, using Gaussian noise). It goes on to mention how, different signal textures or changing cell heterogeneity (different light intensities) can lead to over-segmentation. Also, small pixel distributions and/or irregular cells can impact cell boundary creation. Finally, cell overlap between consecutive frames, and coinciding mitotic events affects tracking.

Ulman et. al. [13] shows the results of competitions held in 2013, 2014 and 2015, through the *IEEE International Symposium on Biomedical Imaging* (ISBI), comparing 21 algorithms. These contained a mixture of traditional and deep learning algorithms. Tables 2 and 3 list the methods used for segmentation and tracking in these competitions.

Since 2017, the competition has been available for online submissions. 2019 and 2020 held additional activities through ISBI, and the 2021 ISBI challenge has recently completed. The results from these years is yet to be published, at the time of writing.

Most state of the art deep learning methods are based on, or inspired by a network called U-Net. This Neural Network (NN) was successful in segmenting a broad range of biomedical images, and is a kind of benchmark to refer to and improve upon. This model is outlined in Ronneberger et. al. [10], and explains how the U-Net design improves upon the Sliding-Window Convolutional Network (A previously successful method). Specifically, there is a contracting left path, and an expansive right path. The left path uses a normal architecture of a CNN, whereas the right path up-samples the results from the contracting steps. Overall, 23 Convolutional layers are used. The benefit of this work is to improve the resolution of the output - and segment the cells efficiently.

It is worth mentioning that Ronneberger et. al. used a 6GB NVidia Titan GPU, and the training took 10 hours. However, the trained models are saved and can be applied to other tasks as well.

Looking closely at existing deep learning architecture, Moen et. al. [8] treats this tracking problem as a linear assignment problem, and uses a supervised deep learning model to optimise a cost function for tracking cells. By using certain cell features, a Hungarian Algorithm can be implemented to select one value - minimizing the cost function and enabling learning.

For segmentation, every cell in a frame is allocated an id and a JSON file, containing lineage information. These annotations are then used in the tracking step. Nuclear seed segmentation was used, and to validate data, crowd-sourcing via a program called *Caliban* enabled users to correct errors with a keyboard and mouse.

To accomplish the tracking: 4 vectors of information (appearance, morphology, motion and neighborhood) were created and given branches. The branches summarize the information provided to it, and then feed the information into the neural network (A hybrid recurrent convolutional deep learning model) To incorporate temporal information, a LSTM layer may merge any multi-frame information into 4 individual vectors.

The neural network has several layers, the final one applying a softmax for a classification: P_{same},

$P_{different}$, $P_{parent-child}$ - probabilities that can then be used for tracking.

Al-Kofahi et. al. [4], on the other hand, uses a deep learning model to segment images. Tracking is not considered by the paper, but the segmentation is successful for cells that have a variety of stains.

The algorithm requires offline training. Thereafter, there are 4 steps to the algorithm: provide the unseen image, separate nuclei and cytoplasm, detect the nuclei, and detect the cell boundaries. The output of the model is 2 images: an image of nuclear seeds (which could be used for tracking) and an image of whole cell boundaries.

The nuclei detection uses a multi-scale blob detector, multi-level thresholding and a shape-based watershed segmentation. The cell segmentation uses pixel-level weighting, multi-level thresholding and a seed watershed algorithm.

5 convolution and pooling paths are used, followed by 2 dropout and de-convolution paths. The paper also provides a list of parameters used, which were proposed as desirable for their architecture.

Scherr et. al. [11] uses deep learning for segmentation, and a graph-based matching strategy for tracking. The paper considers using euclidean distance transform for cell boundaries and the inverse normalized distance for neighbour distances to conduct segmentation. It further mentions how combining cell distances with neighbour distances prevents cellular merging.

For segmentation, 1 encoder path connects to 2 parallel decoder paths, and backpropagation from both decoder branches is allowed. Each branch focusses on 1 task: cell distance or neighbour distance.

For tracking, a graph based cell linking algorithm is used. A rectangular region of interest is used as a search space in each frame, and a phase correlation calculation performed for most likely movement. To connect cells, an adapted version of the coupled minimum-cost flow algorithm is used.

Post-processing for both segmentation and tracking is employed, using a watershed method and mask placement method respectively.

Wang, A et. al. [14] proposes a deep learning based pipeline, focusing on segmenting densely packed 3D cells. Their system uses a 2 stage pipeline and 1 hyper parameter for a lightweight deep CNN model. The model outputs voxel masks. Additional tools are also discussed: a specialized loss function to detect clustered cells, and details on their touching area-based clustering algorithm to partition foreground and background. For post-processing, they propose an algorithm Touching Area-based Spatial Clustering of Applications with Noise (TASCAN) - which is similar to the Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithm.

The hyper parameter is the minimum touching area between 2 voxels in the foreground.
Table 4 is adapted from Wang, A et. al. [14], as it shows an overview of existing deep learning models for cell segmentation.

Ben-Haim and Riklin-Raviv [5] use 2 deep learning tools to tackle tracking: a Multi-Layer Perceptron (MLP) to generate instances and features of each cell, and a Graph Neural Network (GNN) to connect and track cells. Their MLP uses cell segmentation maps or marker annotations to crop each frame into several sub-images - each containing 1 cell. A hard mining strategy and multi-similarity loss function is the used to train a ResNet network.

The paper provides a detailed breakdown of their Deep Learning architecture - and mentions a new GNN block that allows the system to update both the node and edge feature vectors, at the same time. This dual-update system results in a message passing process between nodes. By

designing Message Passing Neural Network (MPNN) blocks, the cell instances and associated feature vectors are encoded as nodes, whereas the movement of cells is encoded as edges. This approach also enables global temporal knowledge, instead of just neighboring frames.

The paper approaches to solve an edge classification problem in their GNN, producing active edges. These active edges are then used to construct the tracks and lineage trees, and can be used for Mitosis detection.

Wang, J et. al. [15] approaches tracking using a Deep Reinforcement Learning (DRL) method to link cells between frames. A cost matrix is created using the cell target features, which is then provided to the network as input. The Reinforcement Learning Neural Network (RLNN) predicts the distribution over a solution. In addition to this, a Residual Convolutional Neural Network (RCNN) is mentioned, to improve the learning rate.

The paper uses a U-Net segmentation method to detect cells in a frame, then a single hypothesis tracking method using a Kalman filter and frame-by-frame association produces the cell movement. The specific architecture is a Residual CNN (ResCNN), made up of 5 blocks. The output of these blocks is a noise value, which is used to calculate a probability distribution.

Fazeli et. al. [6] takes a different approach, and builds a plugin for existing software TrackMate. A pipeline called StarDist is used to ... StarDist is compatible with both fluorescent and widefield images, and has dependencies ZeroCostDL4Mic and Fiji - open-source software. It is designed to be used by biologists, whom have no programming experience, and produce a file containing all the nuclei's geometric center coordinates. These coordinates are then fed into TrackMate.

Pre-Processing	**Principle**	**Feature**	**Methodology**	**Post-Processing**
Noise Suppression	Homogeneity	Intensity	Thresholding	Size Filtering
Intensity Normalization	Homogeneity Boundary	Texture Descriptor	Energy Minimization	Cluster Seperation
Intensity Clipping	Homogeneity Peak	Local Descriptor	Machine Learning	Size Filtering
Illumination Correction			Region Growing	Hole Filling
Image Re-Sampling				Boundary Refinement
Image Sub-Sampling				Region Merging

Table 2: Table showing the different segmentation methods used in previous CTC competitions [13]

Temporal Support	**Principle**	**Division Detection**	**Methodology**	**Post-Processing**
2 Frames	Association	Specific	Graph-Based Multiple-Hypothesis Tracking	Distance-Based Track Refinement
3 Frames	Pre-Processing	Inherent	Graph-Based Shortest-Path Globalization	Cell-Collision-Based Track Refinement
All Frames			Motion-Prediction-Based Propagation	Location-Based Track Refinement
			Maximum-Overlap-Based Propagation	Length-Based Track Refinement
			Overlap-Based Label Propagation	Adjacency-Based Track Refinement
			Constrained Nearest-Neighbour-Linking	Overlap-Based Track Refinement
			Non-Constrained Nearest-Neighbour-Linking	
			Probability Graph-Based Global Optimization	
			Contour Evolution with Motion-Compensation	
			Contour Evolution with Bleaching-Compensation	

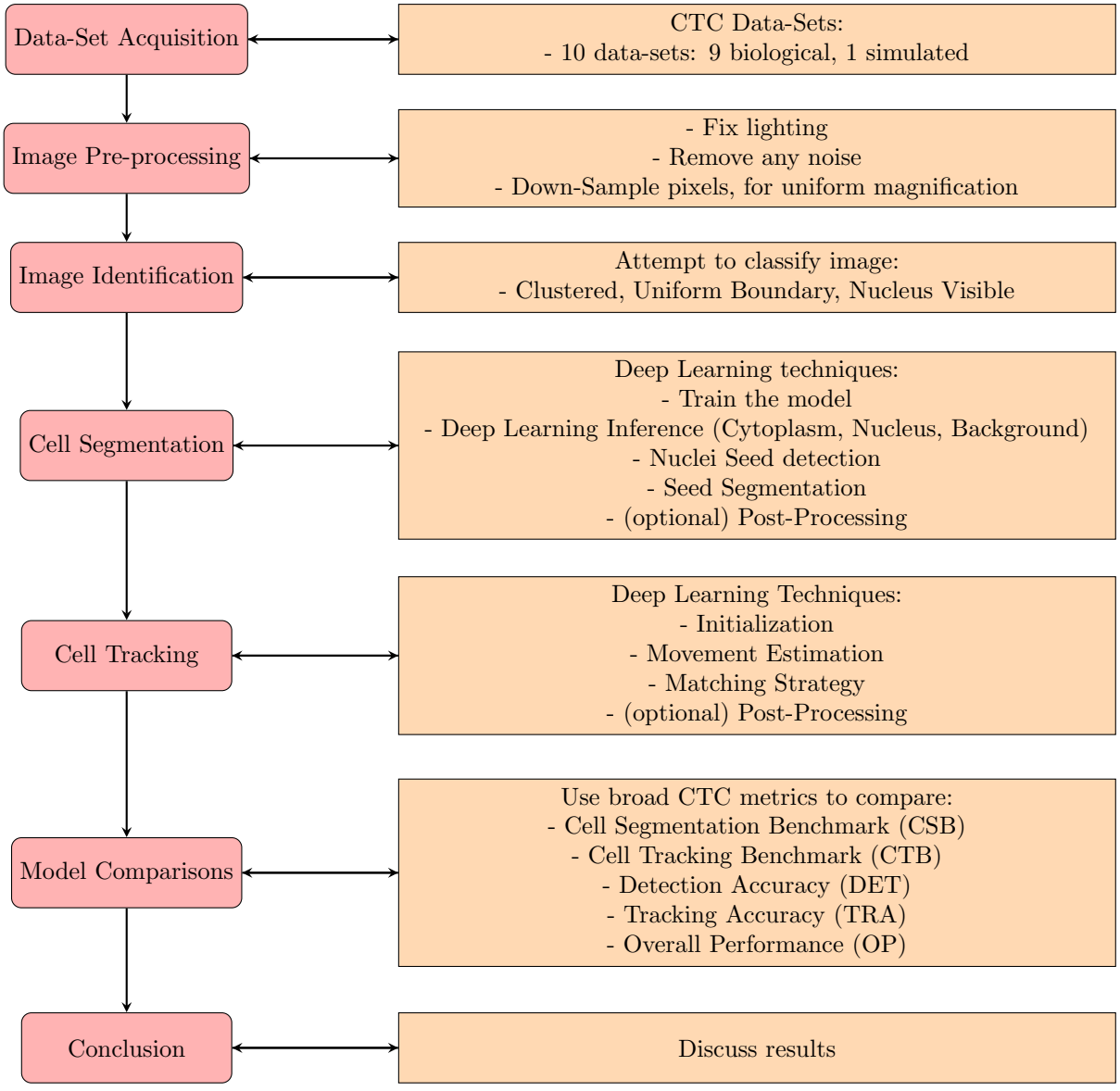
Table 3: Table showing the different tracking methods used in the previous CTC competitions

Segmentation Type		**2D Deep Learning Models**	**3D Deep Learning Models**	**Major Drawbacks**
Semantic Segmentation		U-Net; DeepCell	3D U-net; V-Net; VoxRexNet; 3D-DSN; 2D-3D; C2FNAS; Automatic Data Augmentation	Fails to distinguish different cell instances
Instance Segmentation	Contour-aware approach	DCAN; Deep Watershed	U-Net+CRF; U-Net+SWS; PlantSeg; DISCo; U-Net+Graph-based	Manually selected parameters for post-processing; Prone to fuse tightly packed cells
	Object-detection-based	Retinanet; R-CNN; Keypoint bounding box; PointINS; FCOS; CenterMask; YOLACT	Retina-Unet; Weak Annotation	Imbalance between the number of positive and negative anchor boxes; May fail to discern poorly approximated objects with bounding boxes
	Other strategies	GAN; Embedding; StarDist; TensorMask; AdaptIS; CondInst	StarDist 3D; ShapeMetrics; Spherical Harmonics	Less accurate than the previous two strategies; The training process of GAN networks is highly complex, especially on 3D data-sets

Table 4: Table showing an overview of existing deep learning segmentation methods, adapted from Wang, A et.al. [14]

4 Proposed Methodology and Techniques

The flow diagram below shows the methodology I would like to use:



5 Evaluation and Validation

There are 4 metrics provided by the CTC [1], which will be used. These are shown in the table 5.

The DET metric considers how well each object has been detected, the TRA metric considers how well each object has been tracked and the benchmarks indicate a desirable score, for each data-set.

Broad-Level CTC Metrics	Equation
Cell Segmentation Benchmark (CSB)	*Provided by CTC*
Cell Tracking Benchmark (CTB)	*Provided by CTC*
Detection Accuracy (DET)	$DET = 1 - \frac{min(AOGM-D, AOGM-D_0)}{AOGM-D_0}$
Tracking Accuracy (TRA)	$TRA = 1 - \frac{min(AOGM, AOGM_0)}{AOGM_0}$
Overall Performance (OP)	$OP_{CSB} = \frac{DET+SEG}{2}$, $OP_{CTB} = \frac{SEG+TRA}{2}$

Table 5: Table showing the metrics provided by the CTC [3]

As a result of the CTC competitions in the past, several benchmarks have been found for each data-set.

To aid in calculating the metrics, CTC [3] provides a command-line evaluation software, which calculates DET, SEG and TRA for the user.

The Acyclic Oriented Graph Matching measure for Detection ($AOGM-D$) is used above. $AOGM-D$ is the cost of transforming nodes into Ground Truth (GT), $AOGM-D_0$ is the cost of creating the GT from scratch. SEG is based on the Jaccard similarity index between 2 sets of pixels, trying to match objects.

6 Software Specification

The software for this project will be built in stages. To begin, 4 Jupyter Notebooks will be created. These notebook would be mounted on a high performance PC, and provided with all of the data-sets in a local directory. The following procedure could be used:

- Conduct Pre-Processing on each data-set, and write the results to a new folder
- Use the Pre-Processed images to perform Image Identification (if possible)
- Conduct Cell Segmentation and Tracking
- Compare the models

Thereafter, the working code could be combined into a single file, refined if necessary, and possibly implemented with a GUI.

Python libraries that may be used for Deep Learning include: PyTorch, MXNet, ITK.

Python libraries that may be used for segmentation techniques, and GUI design (in general) include: OpenCV, Tkinter, Scikit.

7 Expected Results and Outputs

From the research done so far, it appears that segmentation is what influences whether the tracking algorithm is successful or not. Most issues experienced at cell tracking level come about because of segmentation errors, though another challenge is cells moving out of frame. Thus, this project will need to explore many segmentation techniques first.

Many research papers have found success using CNN models. In particular, models that are based upon the U-net architecture appear to perform reliably over distinct data-sets.

Ulman et. al. [13] mentions how, as a result of their competition, tracking by detection appears to be more successful than tracking by contours.

8 Possible Extensions

This research could be used to help develop a global cell tracking software. It may also be possible for software to be extended or created to classify a data-set (stem cells, particles, bacteria, etc.)

For example Al-Kofahi et. al. [4] mentions how pixel-based classification is also possible.

Extensions of this research could include: optimizing architecture, introducing additional metrics, lineage tracing, velocity tracing and cell identification.

9 Timeframe

Task	Start Date	End Date	Duration(hours)
Software and Articles	2022/04/02	Current	—
General Research	2022/04/02	2022/06/03	30
Laptop Collection	2022/06/02	—	—
Image acquisition and Pre-processing	2022/06/06	2022/06/17	(predicted) 2 weeks
Image Identification	2022/06/20	2022/06/24	(predicted) 1 week
Cell Segmentation Techniques	2022/06/27	2022/07/15	(predicted) 3 weeks
Cell Tracking Techniques	2022/07/18	2022/08/05	(predicted) 3 weeks
Model Comparisons	2022/08/08	2022/08/19	(predicted) 2 weeks
Software Design	2022/08/22	2022/09/16	(predicted) 4 weeks
Conclusion and Final Write Up	2022/09/19	2022/09/30	(predicted) 2 weeks
End to End Draft Due	2022/09/30	2022/09/30	—
Final Hand-In Due	2022/10/28	2022/10/28	—
Project Presentation	2022/11/10	2022/11/11	—

Table 6: The table presents the proposed timeline

References

[1] Evaluation methodology – cell tracking challenge. URL: http://celltrackingchallenge.net/evaluation-methodology/.

[2] The transmission electron microscope — ccber. URL: https://www.ccber.ucsb.edu/ucsb-natural-history-collections-botanical-plant-anatomy/transmission-electron-microscope.

[3] Cell tracking challenge – where your software moves cells..., 2013. URL: http://celltrackingchallenge.net/.

[4] Yousef Al-Kofahi, Alla Zaltsman, Robert Graves, Will Marshall, and Mirabela Rusu. A deep learning-based algorithm for 2-d cell segmentation in microscopy images. *BMC Bioinformatics*, 19, 10 2018. https://doi.org/10.1186/s12859-018-2375-z doi:10.1186/s12859-018-2375-z.

[5] Tal Ben-Haim and Tammy Riklin-Raviv. Graph neural network for cell tracking in microscopy videos. *arXiv*, 02 2022. URL: https://arxiv.org/abs/2202.04731v1, https://doi.org/10.48550/arXiv.2202.04731 doi:10.48550/arXiv.2202.04731.

[6] Elnaz Fazeli, Nathan H. Roy, Gautier Follain, Romain F. Laine, Lucas von Chamier, Pekka E. Hänninen, John E. Eriksson, Jean-Yves Tinevez, and Guillaume Jacquemet. Automated cell tracking using stardist and trackmate. *F1000Research*, 9:1279, 10 2020. https://doi.org/10.12688/f1000research.27019.1 doi:10.12688/f1000research.27019.1.

[7] Klas EG Magnusson. *Segmentation and Tracking of Cells and Particles in Time-lapse Microscopy*. PhD thesis, 11 2016.

[8] Erick Moen, Enrico Borba, Geneva Miller, Morgan Schwartz, Dylan Bannon, Nora Koe, Isabella Camplisson, Daniel Kyme, Cole Pavelchek, Tyler Price, Takamasa Kudo, Edward Pao, William

Graf, and David Van Valen. Accurate cell tracking and lineage construction in live-cell imaging experiments with deep learning. 10 2019. https://doi.org/10.1101/803205 doi:10.1101/803205.

[9] George Rice. Fluorescent microscopy, 01 2019. URL: https://serc.carleton.edu/microbelife/research$_m ethods/microscopy/fluromic.html$.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science*, pages 234–241, 2015. https://doi.org/10.1007/978-3-319-24574-4$_2 8doi : 10.1007/978 - 3 - 319 - 24574 - 4_2 8$.

[11] Tim Scherr, Katharina Löffler, Moritz Böhland, and Ralf Mikut. Cell segmentation and tracking using cnn-based distance predictions and a graph-based matching strategy. *PLOS ONE*, 15:e0243219, 12 2020. https://doi.org/10.1371/journal.pone.0243219 doi:10.1371/journal.pone.0243219.

[12] Jing Sun, Attila Tárnok, and Xuantao Su. Deep learning-based single-cell optical image studies. *Cytometry Part A*, 97:226–240, 01 2020. https://doi.org/10.1002/cyto.a.23973 doi:10.1002/cyto.a.23973.

[13] Vladimír Ulman, Martin Maška, Klas E G Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, Ihor Smal, Karl Rohr, Joakim Jaldén, Helen M Blau, Oleh Dzyubachyk, Boudewijn Lelieveldt, Pengdong Xiao, Yuexiang Li, Siu-Yeung Cho, Alexandre C Dufour, Jean-Christophe Olivo-Marin, Constantino C Reyes-Aldasoro, Jose A Solis-Lemus, Robert Bensch, Thomas Brox, Johannes Stegmaier, Ralf Mikut, Steffen Wolf, Fred A Hamprecht, Tiago Esteves, Pedro Quelhas, Ömer Demirel, Lars Malmström, Florian Jug, Pavel Tomancak, Erik Meijering, Arrate Muñoz-Barrutia, Michal Kozubek, and Carlos Ortiz-de Solorzano. An objective comparison of cell-tracking algorithms. *Nature Methods*, 14:1141–1152, 10 2017. URL: https://www.nature.com/articles/nmeth.4473, https://doi.org/10.1038/nmeth.4473 doi:10.1038/nmeth.4473.

[14] Andong Wang, Qi Zhang, Yang Han, Sean Megason, Sahand Hormoz, Kishore R. Mosaliganti, Jacqueline C. K. Lam, and Victor O. K. Li. A novel deep learning-based 3d cell segmentation framework for future image-based disease detection. *Scientific Reports*, 12, 01 2022. https://doi.org/10.1038/s41598-021-04048-3 doi:10.1038/s41598-021-04048-3.

[15] Junjie Wang, Xiaohong Su, Lingling Zhao, and Jun Zhang. Deep reinforcement learning for data association in cell tracking. *Frontiers in Bioengineering and Biotechnology*, 8, 04 2020. https://doi.org/10.3389/fbioe.2020.00298 doi:10.3389/fbioe.2020.00298.