COMP9517: Computer Vision

2021 Term 3

Group Project Specification

Maximum Marks Achievable: 40

The group project is **worth 40% of the total course marks**.

Project work is in Weeks 6-10 with a demo and report due in Week 10.

Refer to the separate marking criteria for detailed information on marking.

Submission instructions and a demo schedule will be released later.

Introduction

The goal of the group project is to work together with peers in a team of 4-5 students to solve a computer vision problem and present the solution in both oral and written form.

Each group can meet with their assigned tutor once per week in Weeks 6-9 during the usual consultation session on Thursdays 7-8pm to discuss progress and get feedback.

Description

Tracking of biological cells in time-lapse microscopy images is a common and important computer vision tasks in cell biology [1-3]. To study how cells move, divide, and interact under different conditions (healthy versus diseased), biologists often culture cells in a petri dish and then image them over time using a microscope. The resulting image sequences (videos) are usually too large and contain too many cells to track by hand.

Thus, computer vision methods are needed to automate the segmentation and tracking of the cells, as well as to perform subsequent quantitative analysis of cell motion. Many well-established computer vision methods in conjunction with machine learning methods can be useful in these tasks. These may perform image preprocessing, feature extraction, classification, motion detection, tracking and recognition, using either unsupervised or supervised approaches, including various types of deep neural networks [4-10].

In this project you will develop your own methods for cell segmentation, tracking, and analysis, based on concepts taught in this course or your own ideas.

Tasks

The group project consists of two tasks described below, each of which needs to be completed as a group and will be evaluated for the whole group.

Microscopy Dataset

The image dataset to be used in the group project is provided in WebCMS and consists of four image sequences (each from a separate time-lapse microscopy recording).

For two of the sequences, limited “ground truth” manual annotations are provided (in the corresponding GT folders) in case you want to use/train supervised methods (this is not required but certainly allowed). If more training data would be needed for your methods, you could manually create more annotations yourself.

The ground truth (GT folder) contains both segmentation annotations (SEG subfolder) and tracking annotations (TRA subfolder). The segmentation annotations show cell masks, each with a unique label, for only some of the images in the sequence (the file names of the images indicate the corresponding time points). The tracking annotations show simple cell markers (circles) for all images in the sequence, with a unique label for different cells, but the label is the same for the same cell over time.

Notice that all images are 16-bits/pixel, so make sure to load them as such.

Task 1: Segment and Track Cells

Develop a Python program to segment and track all the cells in the image sequences. This means the program needs to perform the following steps:

1-1. Segment all the cells and show their contours in the images as overlays. The contour of each cell should have a unique color and that color should remain the same for the same cell over time. For each image in a sequence, the program should show the contours for the cells in that image only.

1-2. Track all the cells over time and show their trajectories (also called tracks, paths, histories) as overlays. That is, for each image in a sequence, the program should show for each cell its trajectory up to that time point. The trajectory of a cell is a piecewise linear curve connecting the centroid positions of the cell, from the time when the cell first appeared up to the current time point. For each cell, draw its trajectory using the same color as the contour, for visual consistency. If a cell divides, the two daughter cells should each get a new color/ID.

Task 2: Analyze Cell Motion

Extend the program so that it can quantitatively analyze the cells. Specifically, the program should be able to perform the following analyses. For each image in a sequence, show (by printing either in the terminal window or as overlay in the image window):

2.1. The cell count (the number of cells) in the image.

2.2. The average size (in pixels) of all the cells in the image. To avoid getting distorted measurements, the program needs to ignore cells on the boundary of the image that are not completely contained in the image.

2.3. The average displacement (in pixels) of all the cells, from the previous image to the current image in the sequence. The displacement of a cell can be estimated by taking the Euclidean distance (in pixels) between the centroid positions of the cell (already computed in Task 1-2) from one image to the next. Notice this means for the first image in the sequence, no displacement can be computed.

2.4. The number of cells that are in the process of dividing. Cell division is characterized by a significant change in cell shape (from circular to more bar-like and then resulting in two separate daughter cells changing from bar-like back to circular) and may take multiple time points to complete. In addition to reporting the number of cell divisions, the program should also visually alert the viewer where in the image these divisions are happening (for example, by drawing a thick red circle around dividing cells, or showing some other alert symbol of choice).

Deliverables

The deliverables of the group project are 1) a group video demo with Q&A and 2) a group report. Both are due in Week 10. More detailed information on the two deliverables:

Demo: Each group will prepare a video presentation of at most 10 minutes showing their work. The presentation must start with an introduction of the problem and then explain the used methods, show the obtained results, and discuss these results as well as ideas for future improvements. This part of the presentation should be in the form of a short PowerPoint slideshow. Following this part, the presentation should include a demonstration of the methods/software in action. Of course, some methods may take a long time to compute, so you may record a live demo and then edit it to stay within time.

The entire presentation must be in the form of a video (720p or 1080p mp4 format) of at most 10 minutes (anything beyond that will be cut off). All group members must present (points may be deducted if not), but it is up to you to decide who presents which part (introduction, methods, results, discussion, demonstration). In order for us to verify that all group members are indeed presenting, the head of each student presenting their part must be visible in a corner of the presentation, and when they start presenting, they must mention their name. Overlaying a webcam recording can be easily done using either the video recording functionality of PowerPoint itself (see for example this tutorial) or using other recording software such as OBS Studio, Camtasia, Adobe Premiere, and many others. It is up to you (depending on your preferences and experiences) which software to use, as long as the final video satisfies the requirements mentioned above.

During the scheduled lecture/consultation hours in Week 10, that is Tuesday 16 November 2021 6-8 PM and Thursday 18 November 2021 6-8 PM, the video demos will be shown to the tutors and lecturers, who will mark them and may ask questions about them to the group members. Other students may tune in and ask questions as well. Therefore, all members of each group must be present when their video is shown. A roster will be made and released closer to Week 10, showing when each group is scheduled to present.

Report: Each group will also submit a report (max. 10 pages, 2-column IEEE format) along with the source codes, before 19 November 2021 23:55:00. The report should include:

1. Introduction: Discuss your understanding of the task specification and dataset.
2. Literature Review: Review relevant techniques in literature, along with any necessary background to understand the methods you selected.
3. Methods: Justify and explain the selection of the methods you implemented, using relevant references and theories where necessary.
4. Experimental Results: Explain the experimental setup you used to evaluate the performance of the developed methods and the results you obtained.
5. Discussion: Provide a discussion of the results and method performance, in particular reasons for any failures of the method (if applicable).
6. Conclusion: Summarize what worked / did not work and recommend future work.
7. Contribution of Group Members: State each group member's contribution. In at most 3 lines per member, describe the component(s) each group member contributed to.
8. References: List the literature references and other resources used in your work. All external sources (including websites) used in the project must be referenced.

References

The following papers provide much useful information about microscopic image analysis and cell tracking. If the papers are not directly available (open access) by clicking the links, they should be available online via the UNSW Library.

[1] E. Meijering, O. Dzyubachyk, I. Smal, W. A. van Cappellen. Tracking in cell and developmental biology. *Seminars in Cell and Developmental Biology*, vol. 20, no. 8, pp. 894-902, October 2009. https://doi.org/10.1016/j.semcdb.2009.07.004

[2] C.-M. Svensson et al. Untangling cell tracks: quantifying cell migration by time lapse image data analysis. Cytometry Part A, vol. 93, no. 3, pp. 357-370, March 2018. https://doi.org/10.1002/cyto.a.23249

[3] A.-A. Liu et al. Mitosis detection in phase contrast microscopy image sequences of stem cell populations: a critical review. IEEE Transactions on Big Data, vol. 3, no. 4, pp. 443-457, October 2017. https://doi.org/10.1109/TBDATA.2017.2721438

[4] J. C. Caicedo et al. Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. Cytometry Part A, vol. 95, no. 9, pp. 952-965, September 2019. https://doi.org/10.1002/cyto.a.23863

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

[6] E. Moen et al. Deep learning for cellular image analysis. Nature Methods, vol. 16, no. 12, pp. 1233-1246, December 2019. https://doi.org/10.1038/s41592-019-0403-1

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

[8] X. Lou et al. Active structured learning for cell tracking: algorithm, framework, and usability. IEEE Transactions on Medical Imaging, vol. 33, no. 4, pp. 849-860, April 2014. https://doi.org/10.1109/TMI.2013.2296937

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

[10] V. Ulman et al. An objective comparison of cell-tracking algorithms. Nature Methods, vol. 14, no. 2, pp. 1141-1152, December 2017. https://doi.org/10.1038/nmeth.4473

Released: 15 October 2021