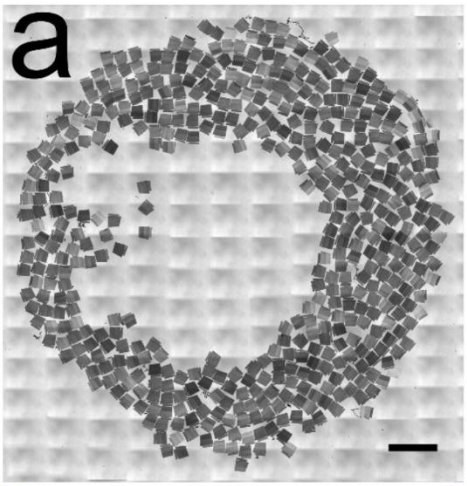
Segmentation of ultrathin sections of biological tissue in light microscopy imagery

Thomas Templier – Interdisciplinary Center for Electron Microscopy (CIME) – thomas.templier@epfl.ch

Machine Learning Student project

A new technology is being developed at the Center for Interdisciplinary Electron Microscopy, EPFL, to collect hundreds to thousands of consecutive ultrathin sections (about 50 nm thicknsess) of plastified biological samples onto silicon wafers in order to perform large-scale light or electron microscopy.



In the new approach, the ultrathin sections, with typical dimensions of 1 mm x 1 mm x 50 nm are deposited in a random fashion onto rigid substrates such as silicon wafers. The goal is to accurately segment each of the deposited sections in a large image to allow coordinates mapping for automated image acquisition in a high resolution microscope such as an electron microscope. It is a standard problem of instance segmentation (the sections are not overlapping between each other).

Each section is made of 2 to 3 parts:

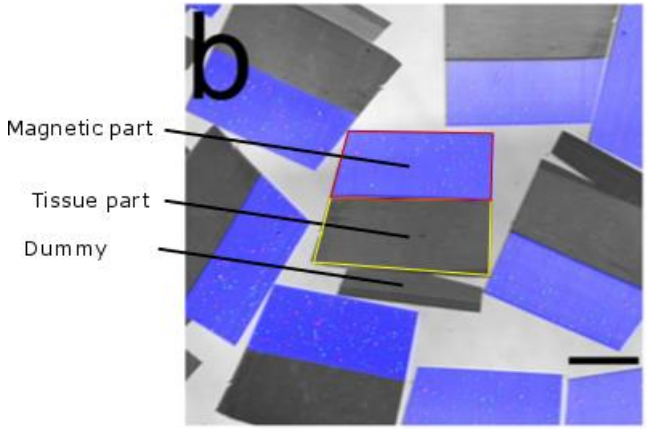
- tissue of interest
- magnetic resin
- possibly a dummy part of tissue

Input

- large 8-bit grayscale image
- manual annotation of a few sections

Output

- center and angle of tissue of interest part
- center and angle of magnetic part (could be simply inferred from position of tissue part)
- user-friendly optimize GUI to correct results



Tasks

- read provided literature about instance segmentation (find more if needed)
- Run Mask-R-CNN (adapt if needed)
- Infer center and orientation of tissue part

Literature:

- https://github.com/matterport/Mask_RCNN : the great tool developed by facebook, state of the art instance segmentation tool
- youtube tutorial for example: https://www.youtube.com/watch?v=2TikTv6PWDw
- https://www.mdpi.com/2072-4292/10/9/1487 : nice use of Mask RCNN, looks similar to the section segmentation problem
- https://github.com/oist/usiigaci : some instance segmentation again, not sure how useful it is
- https://medium.com/geoai/parking-lot-vehicle-detection-using-deep-learning-49597917bc4a : detecting vehicles
- https://github.com/jremillard/images-to-osm : another use of mask rcnn
- easy to use GUI to manually create polygons: https://github.com/wkentaro/labelme

The tree main parts of a ML pipeline:

1. Efficient ground truth generation

Use and potentially enhance the software labelme to generate ground truth rapidly.

Is ground truth generation tedious ? Set up a mechanical turk on Amazon !
http://labelme.csail.mit.edu/Release3.0/browserTools/php/mechanical_turk.php

2. ML inference

The core that generates predictions. Is it possible to label a dozen of objects in a new dataset and to rapidly get the prediction, so that the user can see whether there is enough ground truth ?

3. Proofreading

The overlooked part of ML: proofreading is very common. Design the most efficient way to proofread the predictions. Is it too tedious ? Set up a mechanical turk again !