Programming Test to Evaluate Candidate

Synthetic Dataset for Cell Segmentation

⚠ Confidential material of St-Pierre Lab @ Baylor College of Medicine.

Background

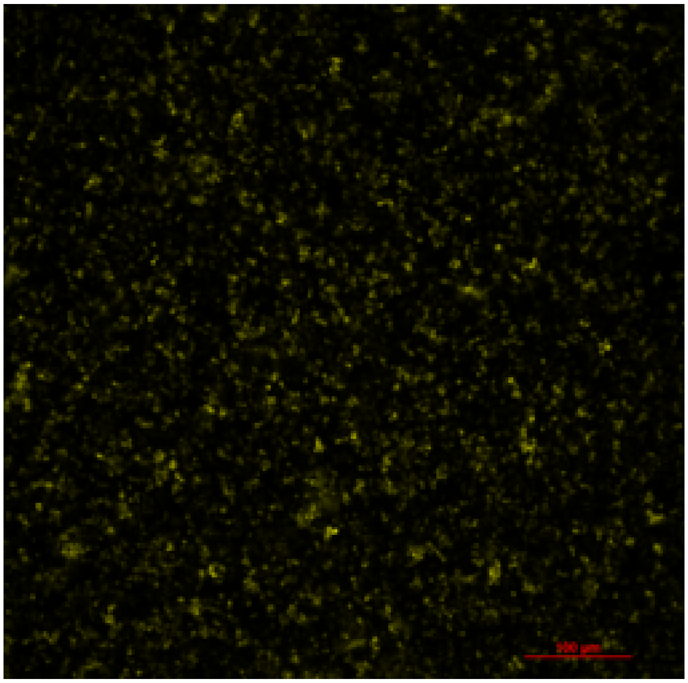
Image segmentation is widely used in analysis of microscopy images of cells as a prerequisite for quantification of many visual phenotypes. Image segmentation can be accomplished using many computer vision and image processing techniques, depending on the goal of the segmentation. In some supervised techniques, such as deep learning, a training dataset is needed to tune the model. This dataset is commonly obtained by manually labeling the images. For example, to create an instance segmenter that can pick out pedestrians from street camera imagery, actual images from the scene and their corresponding labeled images are provided to train the algorithm. The labeled images have pixels corresponding to the pedestrians manually marked 1, and the rest marked as 0. Given the training process can take thousands of these image /label pairs, the process of curating a training dataset is understandably labor intensive.

However, there is one additional challenge when the ground truth is hard to determine, as in the example above, when the scene is dark and even our eyes can hardly pick out pedestrians. This can be overcomed by having multiple people labeling the same image and get their average response. You can see that this further lengthens the process. An alternative to completely bypass the need of inferring the correct label from the image is to eliminate the inference in the first place. This is by implementing a **synthetic image generator** to create real-life looking images directly from input labels.

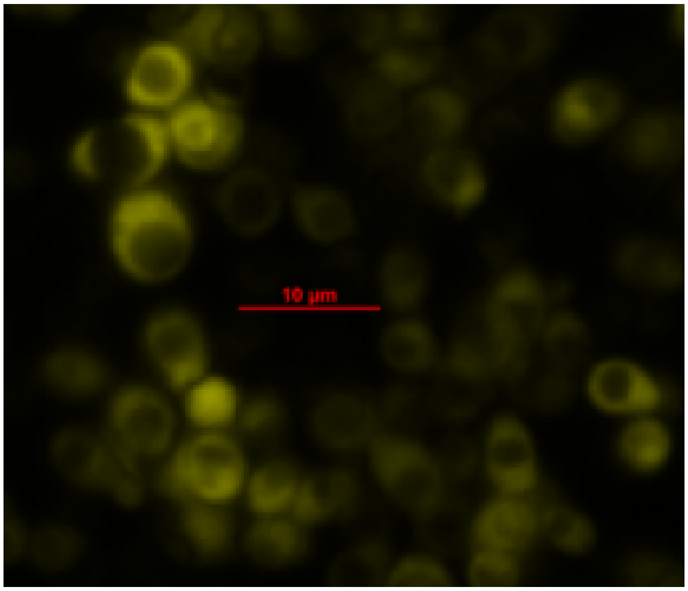
In our lab, we are interested in developing protein tools for neuroscience applications. One of the steps involves evaluating the fluorescence signals of cells given by the protein indicators we are engineering. For that, we need a good instance segmentation algorithm of cells (yeast, mammalian cells, etc.) to distinguish one cell from another before quantifying their fluorescence response. As mentioned above, a training dataset could help us create models that can be used for segmentation, but as you can see, it is really hard and tedious to create a ground truth labeled image for an image that easily contains thousands of cells.

Task

Your task is to create a **synthetic image generator** to generate image and labeled image pairs of yeast cells under fluorescence microscopy. Here is a representative fluorescence image we captured in our lab for your reference.



The image is 2048 x 2048 pixels and is monochromatic. It is displayed in false color in yellow. When we zoom in, we can see individual yeast cells.



The main function of the synthetic image generator accepts **inputs** as follows:

1. Property of the image: width, height.

2. Features of the cells: number of cells, fluorescence level, size, shape, location, etc.

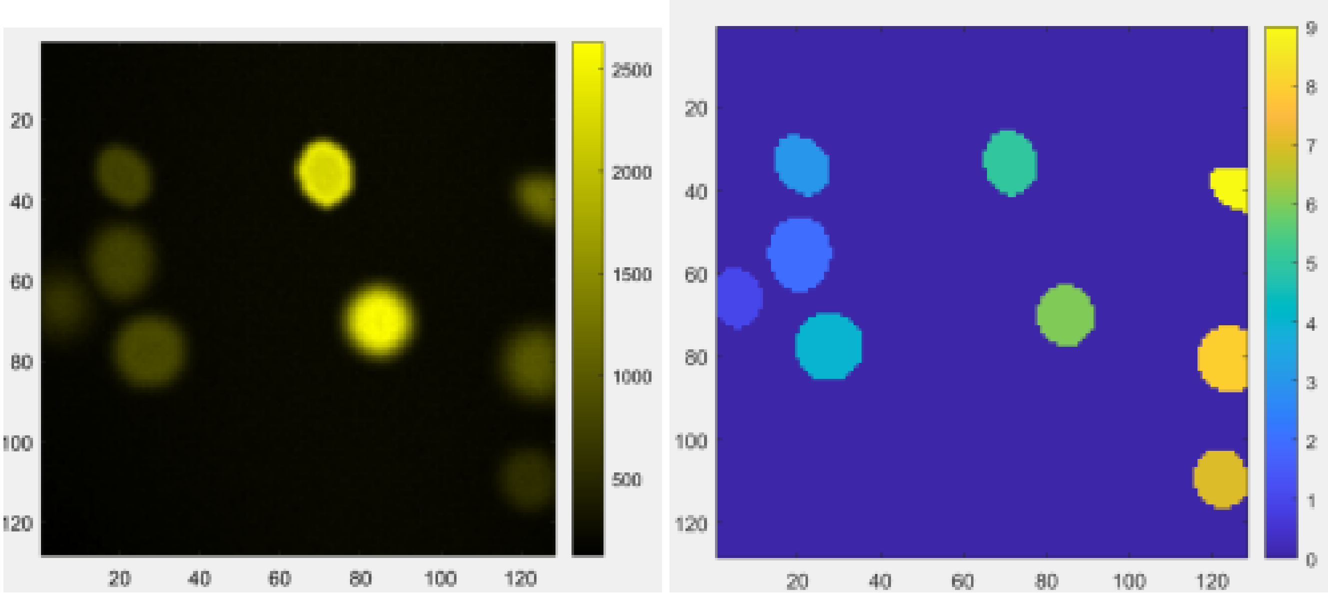
3. Additional properties that might affect how the fluorescence image would look like. Be reasonable and make it as realistic as you can (hint: consider variation of cells and noise from camera), but you are not required to implement all permutation of inputs, i.e. you may tell us that certain arguments only work for a collection of values.

The **outputs** should be:

1. The generated fluorescence image of yeasts in uint16.

2. The corresponding labeled image in uint8 (background is 0, and cells are labeled in increment numbers, uint8 is assuming the maximum supported cell count per image is 255).

Here is an example output when we requested an image of 128 x 128 pixels that consists of 9 cells (left is the actual image in fluorescence, and right is the instance segmentation labeled image).



Requirements:

1. Create a **Github** repository to work on the problem.

2. You may use a programming language of your preference, but **MATLAB** is preferred.

3. Clearly **comment** on your code. This includes the meaning of input and output of all functions, important assumptions used, etc.

4. Create a **readme** file. Put any additional observations there. In addition, include an **exemplar script** to demonstrate how to use your code, with exemplar input and expected output.

5. Feel free to use existing algorithms but please list all external references and source code and properly **cite** them in the readme file.

6. We value **correctness**/functionality more than performance but you are welcome to optimize the code for performance given time.

7. You have **1 week** of time to finish the task. If you finish early, you may submit early or consider doing the bonus features below. If you run out of time, just submit the repository as is and no extension will be given.

8. Don't hesitate to ask us any questions on this project. You may send emails to Haixin Liu (haixin.liu@bcm.edu).

Bonus:

If you want to impress us and if you have time, you may consider doing the following:

1. Create a UI that allows people to set the parameters interactively and preview the image/label pair before a larger number of them are generated in one batch.

2. Implement and test an segmentation algorithm (e.g. Mask R-CNN) to see if the generated training set helps training the model.

Deliverables:

Please submit your result as an email to Haixin Liu (haixin.liu@bcm.edu). In the email, please include:

1. The link to the Github repository that contains the files (readme, source code, documentation etc.);

2. A paragraph of 'your feeling' when doing this task. For example, how challenging or easy do you think it is? Did you try some methods that didn't work? How satisfied are you with the result? How would you do it better if you were given more time?