

Medical Image Processing

Instruction Manual

ECE 435 / BME 403



**University
of Victoria**

Kimiya Mousavi - V00919059

Miguel Manguerra - V00917855

Due date: April 2nd, 2023

Function Description

There are 3 separate functions in separate files for this project that are all part of the detection and segmentation algorithm. The files are:

1. `segmentNuclei.m`

This function takes in the path to the image that you want to segment, value *m* which is the max size of the nuclei to segment, value *s* which is the solidity of the nuclei you want to segment (between 0-1), *t1* & *t2* which are the min and max intensity values that the function will segment (between 0-255) and value *d* which is the boundary intensity difference.

The output of the function is the nucleus mask (a binary image) that highlights the nuclei of the image. It will also display a figure that compares the original image to the output binary nucleus mask highlighting where each nucleus is.

This is a function call to segment nuclei that will lead to a very good nucleus segmentation of EDF000.png:

```
nucleusMask = segmentNuclei("EDF000.png", 50, 0.95, 10, 210, 2.5);
```

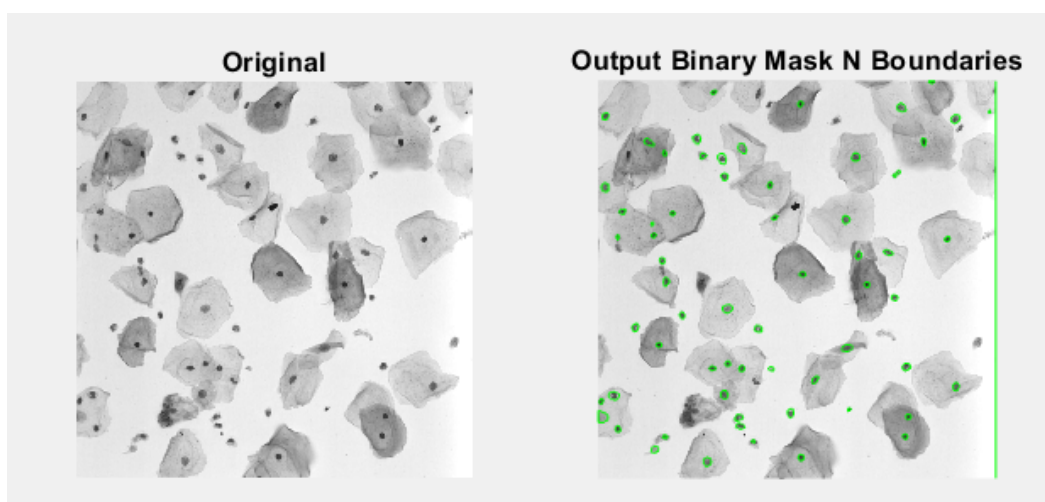


Figure 1 - Output of running the given function call for `segmentNuclei`

2. `segmentClumps.m`

This function takes in the path to the image you want to segment, the `nucleusMask` (output of `segmentNuclei` for the same image) and the minimum area of each clump. The output of this function is the image with only the clumps cells included. It will also display a figure that compares the original image to the segmented clumps output which draws a border around each cell clump in the image.

A good call of this function that returns well segmented clumps is the following:

```
segmentedClumps = segmentClumps('EDF000.png', nucleusMask, 1000);
```

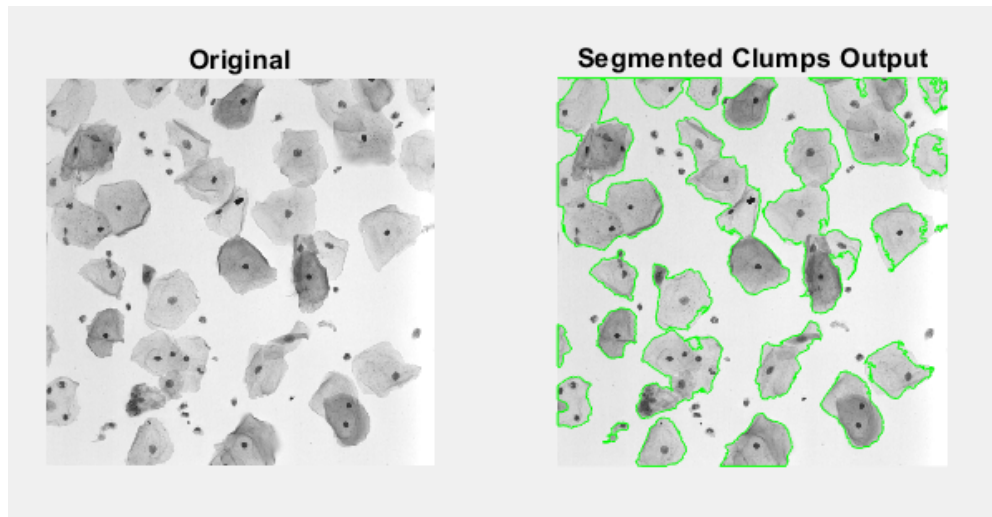


Figure 2 - Output of running the given function call for segmentClumps

3. segmentCytoplasm.m

This function takes in the path to the input image you want to segment, the nucleusMask (output of segmentNuclei for the same image), the segmentedClumps (output of segmentClumps for the same image), W which is the width of the square grid overlaid over each nuclei (approximate size of each cell), alpha which is the likelihood multiplier of grid squares belonging to the same cell and beta which is the allowed degree of overlap between cells in a cell clump. The function returns a cytoplasmMask which is supposed to outline each cell in the clumps.

This function was not finished, so currently when the following function call is ran:

```
segmentCytoplasm('EDF000.png', nucleusMask, segmentedClumps, 200, 2, 0);
```

The output is a figure that shows an image of the centre of each nuclei in the image and has boxes or width W drawn around each nuclei showing where each cell is in the image.

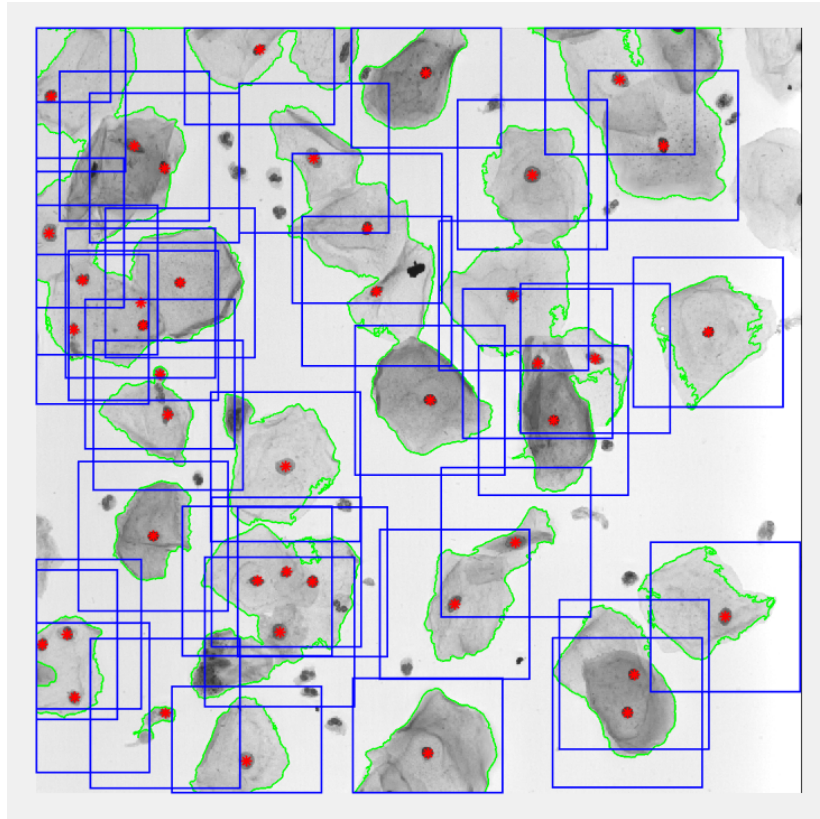


Figure 3 - Output of running the given function call for segmentCytoplasm as shown above

In addition to the three functions that are included in the algorithm, there is also a function called **countNuclei** that compares the ground truth image to the segmented nuclei image and counts the amount of nuclei found within each image. Its purpose was for us to generate quantitative data on the accuracy of our segment nuclei function. The function can be run with the following statement and will display the following output.

```
countNuclei(nucleusMask, 'EDF000_GT.png');
```

```
Nuclei Mask: 65
Ground Truth: 37
```

Figure 4 - Output of countNuclei for EDF000.png

How to run the project

There are 3 separate functions in separate files for this project. To run this project, first run the segmentNuclei function as shown. The runtime for this should be about a minute.

```
nucleusMask = segmentNuclei("EDF000.png", 50, 0.95, 10, 210, 2.5);
```

Next, this same nucleus mask is inputted into the function call for segmentClumps as shown. The runtime for this will be almost immediate.

```
segmentedClumps = segmentClumps('EDF000.png', nucleusMask, 1000);
```

Finally, run `segmentCytoplasm` with the following function call. You must include the `nucleusMask` and `segmentedClumps` output that were found for the same image with the function calls above. Do not try to set a variable equal to this function call, because this function does not currently return anything.

```
segmentCytoplasm('EDF000.png', nucleusMask, segmentedClumps, 200, 2, 0);
```

NOTE: Every function call must have the path to the same image. Also, make sure these function are being called in the “MATLAB code” folder of the Image-Processing-Project.