Biomedical Image Analysis

Final Project Report

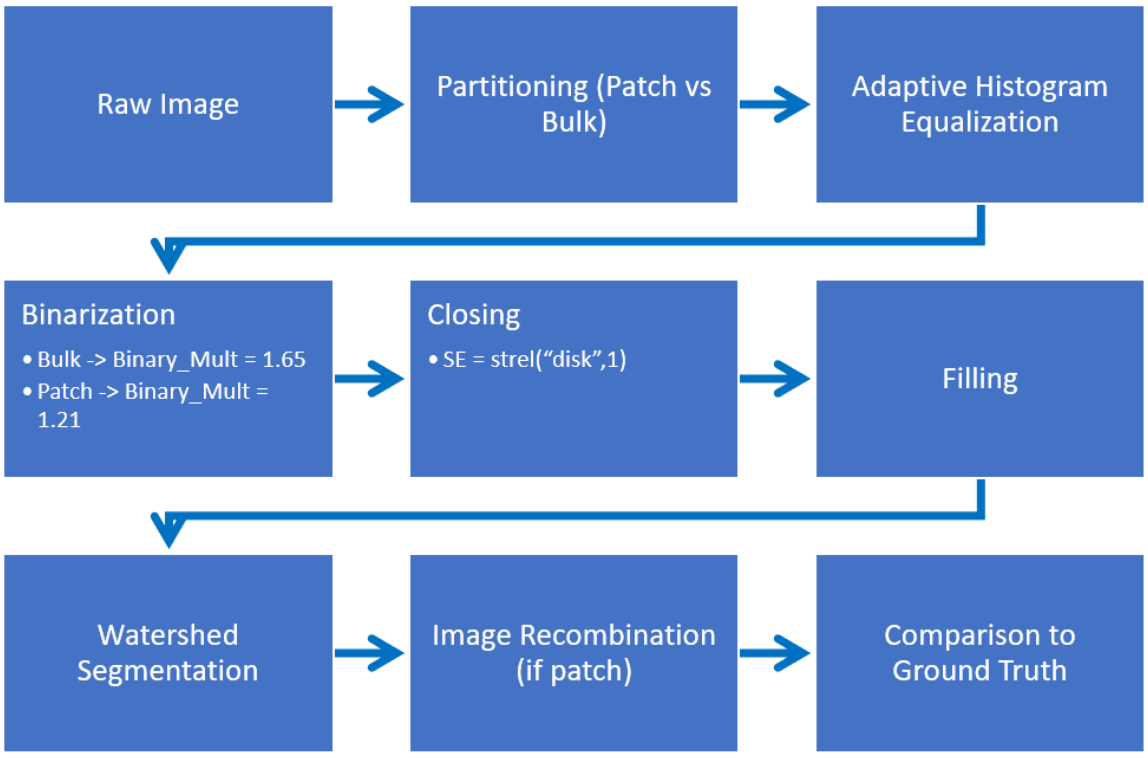
DAPI Segmentation

Parvathy Unnikrishnan, Christian Garcia, Daniel Giraldo

Problem of Interest

In the field of digital and computational pathology, it is often useful to extract features and data from a histology image so that it can be used to assist physicians in diagnosing and/or treating patients. Some of the features that are relevant and useful in determining tissue health come from the nuclei of the cells. Given a DAPI-stained histology image, our group sought to segment the cell nuclei contained in the image utilizing traditional image processing techniques (non-AI techniques). From our resulting segmentation mask, the cell nuclei in the histology image are localized and can be further processed and analyzed by the other groups in the team to extract features.

Method



Code

```
%% Image Partitioning
x_cutoffs = 1:partition_x:size(img,2);
y_cutoffs = 1:partition_y:size(img,1);

if x_cutoffs(end) ~= size(img,2)
    x_cutoffs = [x_cutoffs,size(img,2)+1];
else
end

if y_cutoffs(end) ~= size(img,1)
    y_cutoffs = [y_cutoffs,size(img,1)+1];
else
end

img_table = {};
for ii = 1:length(x_cutoffs)-1
    for jj = 1:length(y_cutoffs)-1
        x_dims = x_cutoffs(ii):x_cutoffs(ii+1)-1;
        y_dims = y_cutoffs(jj):y_cutoffs(jj+1)-1;
        img_table(jj,ii) = {img(y_dims,x_dims)}; %#ok<*AGROW>
    end
end

```

```
%% Adaptive Histogram Equalization
adapt_table = {};
for ii = 1:size(img_table,1)
    for jj = 1:size(img_table,2)
        adapt_table(ii,jj) = {adapthisteq(img_table{ii,jj})};
    end
end

```

```
%% Binarization
all_thresh=[];
binary_table = {};
for ii = 1:size(img_table,1)
    thresh=[];
    for jj = 1:size(img_table,2)
        gray_threshold = graythresh(adapt_table{ii,1})*binary_mult;
%         gray_threshold=0.35;
         if gray_threshold<0.2
             gray_threshold=0.25;
         end
%          gray_threshold=0.26;


        thresh=[thresh,gray_threshold];
        binary_table(ii,jj) = {imbinarize(adapt_table{ii,jj},gray_threshold)};
    end
    all_thresh=[all_thresh;thresh];
end

```

```
%% Closing
open_table  = {};
for ii = 1:size(img_table,1)
    for jj = 1:size(img_table,2)
        open_table(ii,jj) = {imclose(binary_table{ii,jj},se)};
    end
end
```

```
%% Filling
fill_table = {};
for ii = 1:size(img_table,1)
    for jj = 1:size(img_table,2)
        fill_table(ii,jj) = {imfill(open_table{ii,jj},"holes")};
    end
end

%% Watershed Segmentation
watershed_table = {};
all_label={};
for ii = 1:size(img_table,1)
    for jj = 1:size(img_table,2)
        BW_img = fill_table{ii,jj};
        D = ~bwdist(BW_img);
        D = -D;
        label = watershed(D,8);
        label(~BW_img) = 0;
        all_label(ii,jj)={label};
        rgb = label2rgb(label,'jet',[0.5 0.5 0.5]);
        watershed_table(ii,jj) = {rgb};
    end
end

```

Results

Our performance was tested with respect to a ground truth image provided. The performance metrics used to analyze performance were precision, recall, specificity, and F1 score. Our best performance which was produced through the patch segmentation method is as follows:

Precision: 0.7281

Recall: 0.7697

Specificity: 0.9710

F1 Score: 0.7483

Discussion

Overall, our group was satisfied with the performance of our nuclei segmentation algorithm, especially considering only traditional image processing techniques were utilized. It is likely that better performance can be achieved with AI techniques such as deep learning.

One of the challenges we faced was with watershed thresholding. Due to its simple nature, some combinations of overlapped cells were incorrectly segmented from one another, ultimately leading to poor boundaries being chosen to divide cells. It was said in class that this might've been overcome with more advanced methods which made use of machine learning, but this was outside the scope of our project as it currently stands. To that end, we believe that segmentation of overlapped cells is done to the best degree of quality given the tools we have to work with.

Our group also decided to test two different segmentation philosophies. One of them included performing image processing techniques on the entire whole slide image and the other included splitting the image into 256x256 patches and performing image processing techniques on each of the patches. We were able to rectify the patch method, and the results obtained using this segmentation tactic performed slightly better than the bulk segmentation.

Since we were given a ground truth image to compare against we were also able to test a multitude of threshold values in a for loop to find the most optimal value for this threshold. Contrary to our initial belief, the graythresh value obtained via Otsu's method was not the most ideal value. Instead we had to apply a multiplier to this value to get a satisfactory threshold. Upon testing, we discovered that a multiplier of 1.21 worked best with the patch segmentation and that of 1.65 performed well on the bulk segmentation strategy. We also measured the optimal performance achieved by hardcoding the threshold value altogether and found that by setting this value to 0.26 we were able to achieve an F1 score of 0.75. However, for our final workflow we decided to not rely on this hardcoded threshold as it might not be a generalisable parameter.