Table of Contents

read the image

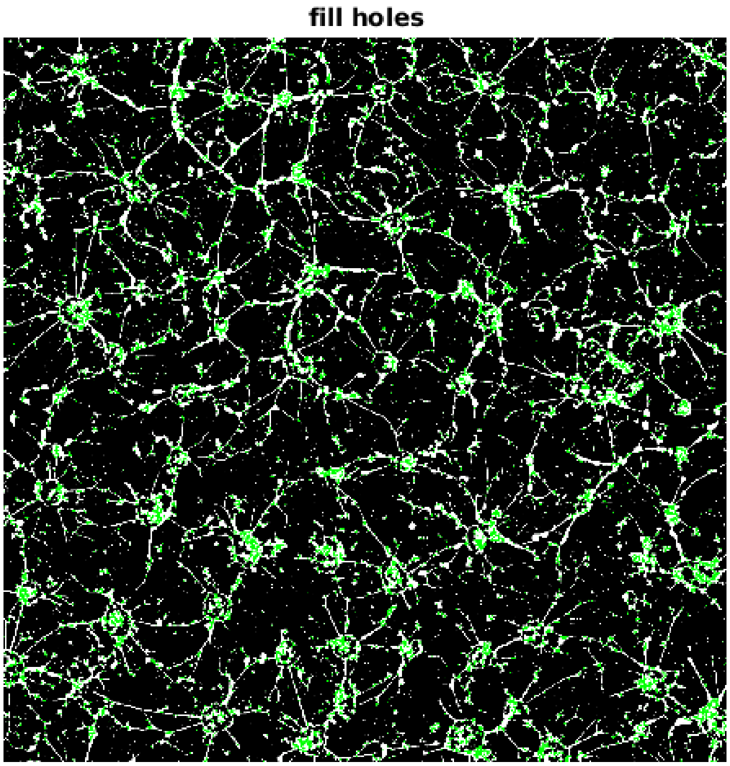
```
img = imread('resized/preprocessed-t0.tif');

image = img(3001:5000, 6001:8000);
```

preprocess

binarize

```
BW_p = image > 40;

% dilate and erode to connect parts of neurons
se = strel('disk', 4);
BW_p = imclose(BW_p, se);

% fill holes in neurons which caused when removing dots
BW_filled = imfill(BW_p, 'holes');
fill_differ = BW_filled & ~BW_p;
% preventing from fill very large holes
min_area = 200;
fill = fill_differ & ~filterRegions_area(fill_differ, min_area);

BW_p = BW_p | fill;
figure;imshowpair(BW_p, image > 40);
title('fill holes');
```

Warning: Image is too big to fit on screen; displaying at 25%



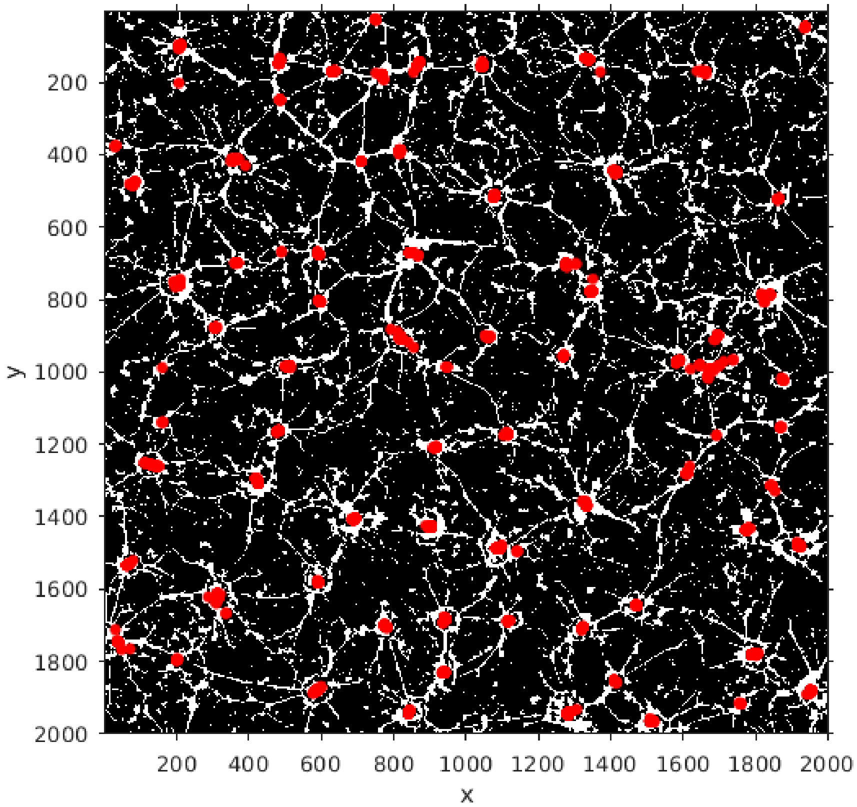
potential neurons

radius range

```
R_center = 65;
R_range = 35;
Neurons = zeros(0, 2);

parfor r = R_center - R_range:R_center + R_range
    % circular kernel: pixels inside circle(r) are 1, outside ones are
 0.
    kernel = circle_kernel(r, 0);

    % conv2
    density = conv2(double(BW_p), kernel, 'same');

    % pixels where conv2 result are larger, which means bright areas
    neu = density > 0.45 * sum(kernel(:));

    % find centroids of these bright areas
    stats = regionprops(neu, 'Centroid');
    len = length(stats);
    Centroids = zeros(len, 2);
    for i = 1:len
        Centroids(i, :) = stats(i).Centroid;
```

```
    end
    Neurons = [Neurons; Centroids];
end

% centroids of bright areas are potential neurons
draw_neurons(BW_p, Neurons);
```

Warning: Image is too big to fit on screen; displaying at 25%



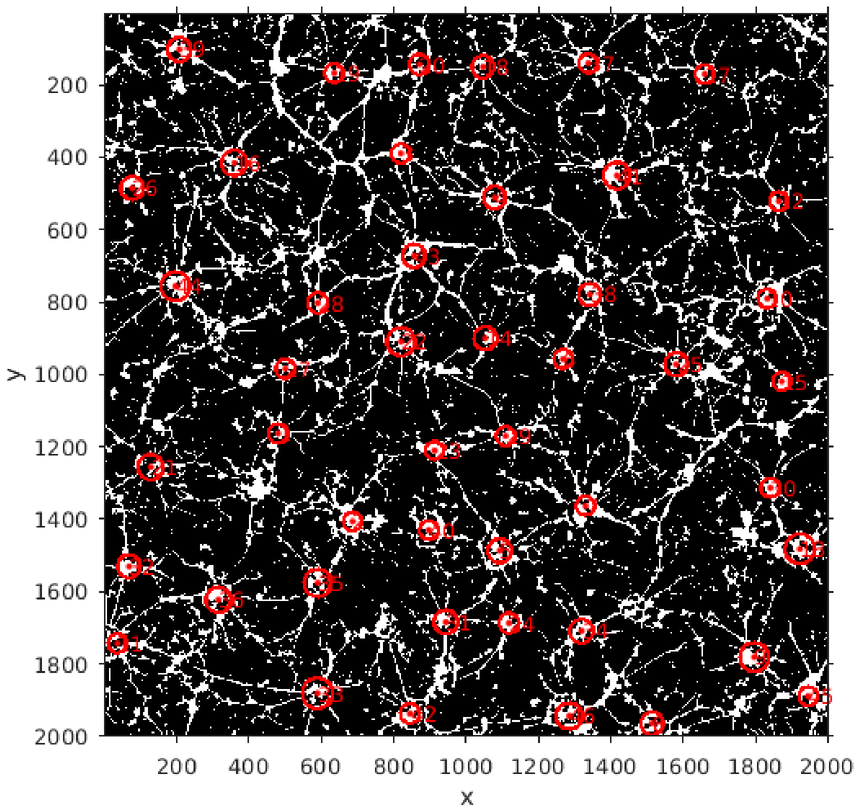
get neurons from centroids

This section tries to select neurons from points we got from lines. The main idea is to calculate numbers of bright angles in an anulus. For each degree x in 0:360, if there are bright pixels in the anulus whose angles from the center of the anulus equal to x,we say x is a bright degree. For a point p, A(p,R,annulus) draws a anulus around p with radius = R and width = annulus * 2. If number of bright degrees in A(p,R,annulus) is more than 360 * threshold_angle, we say point p is a neuron and R is the neuron's size. For every point, we try R in dimension [R - R_range, R + R_range] (that is 45:105, if using parameters below).

```
% Besides, we use another R_around to determine whether there is a
 bright
% area around the center of neuron, because most neurons are bright in
 the
% center.

threshold_angle = 0.55;
R_center = 70;
```

```
R_range = 45;
R_around = 25;
threshold_around = 0.25;

[final_Neurons, grades, R, around] = IsNeurons_new_4(BW_p,
 Neurons, ...
                    'threshold_angle', threshold_angle, ...
                    'merge_dis', 2, ...
                    'R', R_center, 'R_range', R_range, 'annulus',
 2, ...
                    'R_around', R_around, 'threshold_around',
 threshold_around);

draw_circles(final_Neurons, R, BW_p);

% number neurons
for i = 1:length(grades)

 text(final_Neurons(i,1),final_Neurons(i,2),int2str(i),'FontSize',10,'Color','red'
end
```

Warning: Image is too big to fit on screen; displaying at 25%



Published with MATLAB® R2018a