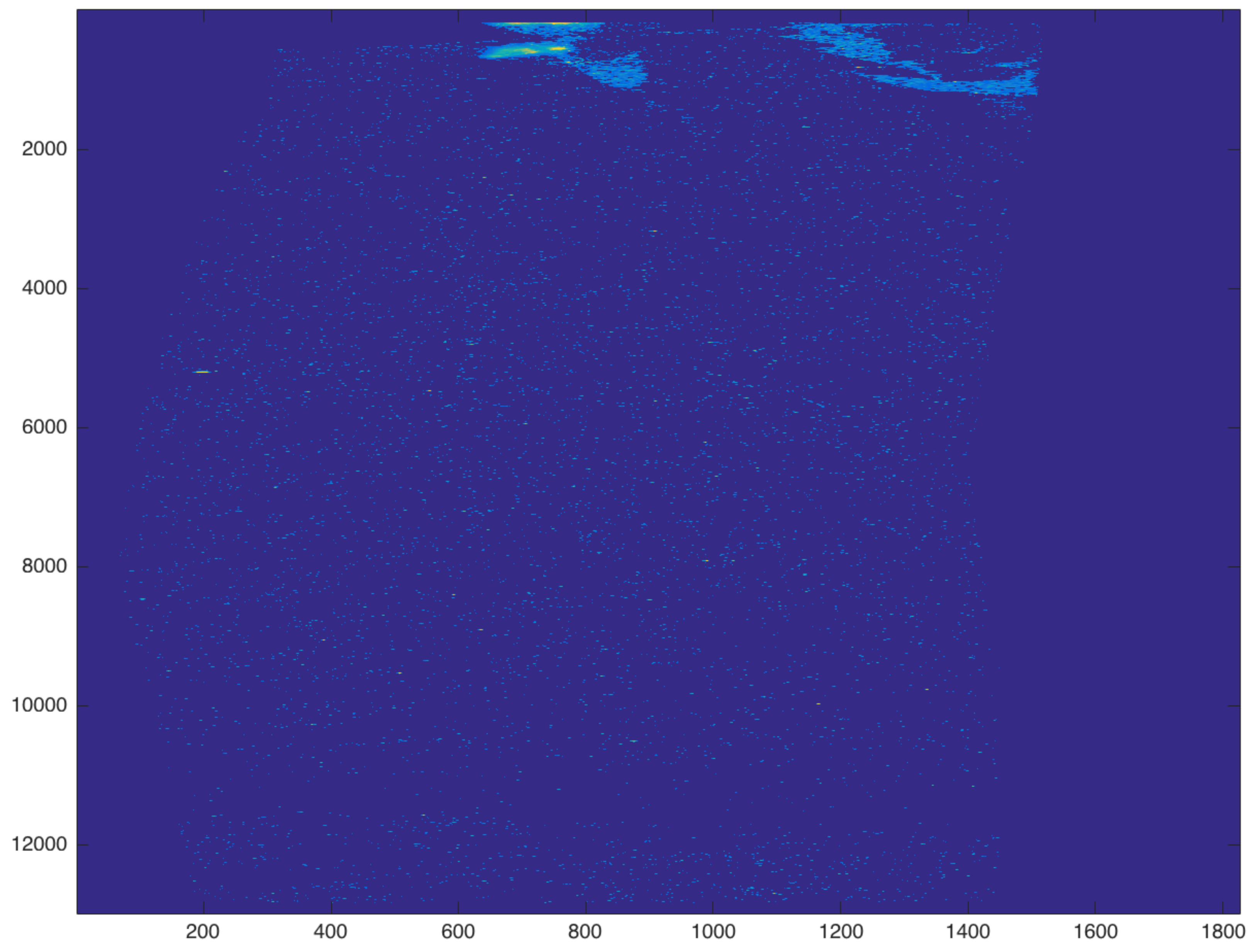
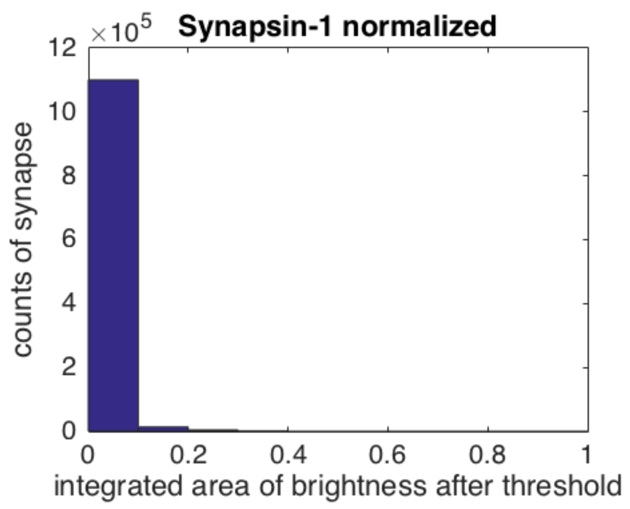
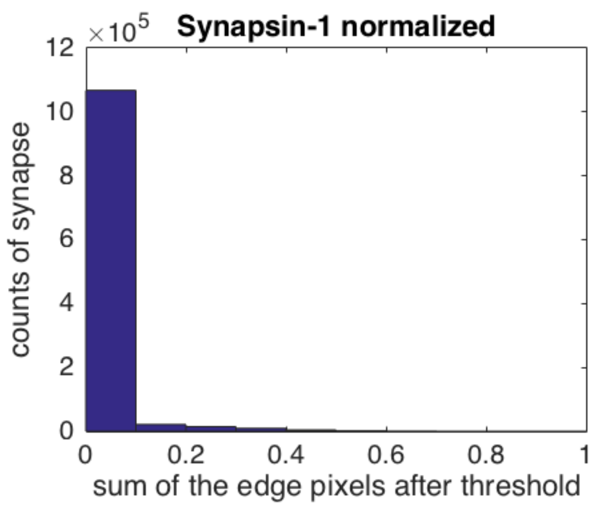
```matlab
% for evaluating the synapse singal strength for each synapse position
% given in the pviot file
system('caffeinate -dims &');
stacks = {'-01-synapsinR_7thA.tif', '-02-synapsinGP_5thA.tif'};

%read in the positions
position = containers.Map('KeyType','char', 'ValueType','any');
fid = fopen('synapsinR_7thA.tif.Pivots-3.txt');
tline = fgetl(fid);
key = '50';
temp = [];

while ischar(tline)

    entry = strsplit(tline,',');
    val1 = entry(1); val2 = entry(2);
    val1 = val1{1}; val2 = val2{1};
    prev_key = key;
    key = entry(3);
    key = key{1};
    if ~isKey(position, key)

        if str2num(key) ~= 50
            position(prev_key) = temp;
        end


        temp = [str2double(val1), str2double(val2)];
    else


        temp = [temp; str2double(val1), str2double(val2)];

    end

    tline = fgetl(fid);

end

position(key) = temp;
fclose(fid);


%define for each posititition a quality metric based on area of the synapse
%or the sum of the edge pixel of the synapse
for image = 1 : 2

    file = stacks{image};

    for k = 1 : 41

        if ~isKey(position, num2str(k))
            continue
        end

        %threshold the image
        curr_position = position(num2str(k));
        [X,map] = imread(file,k);
        X(X < median(X(:))) = 0;
        X ( X <= mean(X(X>0)) + std2(X(X>0)) * 1.7) = 0;
        Area = zeros(numel(curr_position(:, 1)), 3);

        % or do edge detection, and use that as the metric
        Y = edge(X, 'Canny');
        figure(2)
        title('sample image after threshold')
        imagesc(X)


        for i = 1 : numel(curr_position(:, 1))

            entry = curr_position(i, :);

            %look at the neighborhood of the called synapse point, deduce
            %the size of the synapse
            upperx = entry(2) + 8;
            if upperx > size(X, 2)
                upperx = size(X,2);
            end
              lowerx = entry(2) - 8;
            if lowerx < 1
                lowerx = 1;
            end
            uppery = entry(1) + 8;
            if uppery > size(X, 1)
                uppery = size(X,1);
            end
              lowery = entry(1) - 8;
            if lowery < 1
                lowery = 1;
            end

            %set the size of the neighborhood
            integrate = X(lowery : uppery, lowerx : upperx);
            edge_sum = Y(lowery : uppery, lowerx : upperx);
            [columnsInImage, rowsInImage] = meshgrid(1:size(integrate,2), 1:size(integrate, 1));
            circlePixels = (int8(rowsInImage) -idivide(int8(size(integrate,1)), 2, 'floor')) .^2 +
  (int8(columnsInImage) -idivide(int8(size(integrate,2)), 2, 'floor')).^2 <= 15.^2;
            integrate(~circlePixels) = 0;
            edge_sum(~circlePixels) = 0;
            integrate(integrate > 0) = 1;

            %calculaet the area sum and the edge sum
            area = sum(sum(integrate));
            ed = sum(sum(edge_sum));

            Area(i, 1) =  entry(1);
            Area(i, 2) = entry(2);
            Area(i, 3) = area;
            Area(i, 4) = ed;

        end
        if image == 2
        position(num2str(k)) = [position(num2str(k)); Area];
        else
              position(num2str(k)) = Area;
        end


    end

end


temp1 = [];
temp2 = [];
for k = 1 : 41
    if isKey(position, num2str(k))
        temp = position(num2str(k));
        temp1 = [ temp1; temp(1: size(temp, 1) / 2, :)];
        temp2 = [ temp2; temp(size(temp, 1) / 2 + 1 : size(temp, 1), :)];

    end
end

%normalize to 0 to 1
syn1 = [];
syn2 = [];
syn3 = [];
syn4 = [];
syn1 = temp1(:,3) / max(temp1(:,3));
syn2 = temp2(:,3)/ max(temp2(:,3));
syn3 = temp1(:,4) / max(temp1(:,4));
syn4 = temp2(:,4)/ max(temp2(:,4));

disp(sprintf('We try to look at the actual synapsins channel image to get rid of the mislabeled sy
napse.\n It was not obvious as to how we can define a local bright spot on the image as qualified
candidate or not.\n As a result we tried to accomplish this feat by first thresholding the image a
nd then subjecting to the neighborhood of each labeled synapse in the pivot file to two different
metric.\n One based on the number of non-zero pixel in the neighborhod, and one based on the numbe
r of edge pixel in the neighbood.\n For the non-zero pixel metrix, it turned out that for most lab
eled synapse a value of 0 was obtained. We will look into better ways of quantifying this problem,
 but for now we just keep all the synapse position\n with values above a 0. '))
figure(1)
subplot(2,2,1);
xlim([0, 1]);
hist(syn1);
title('Synapsin-1 normalized')
xlabel('integrated area of brightness after threshold')
ylabel('counts of synapse')


subplot(2,2,2);
xlim([0, 1]);
hist(syn3);
title('Synapsin-1 normalized')
xlabel('sum of the edge pixels after threshold')
ylabel('counts of synapse')

subplot(2,2,3);
hist(syn2);
title('Synapsin-2 normalized')
xlabel('integrated area of brightness after threshold')
ylabel('counts of synapse')
```

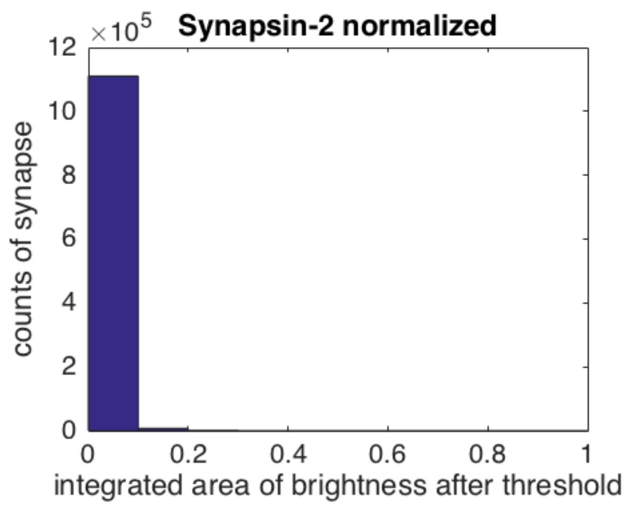
```
subplot(2, 2, 4);
xlim([0, 1]);
hist(syn4);
title('Synapsin-2  normalized')
xlabel('sum of the edge pixels after threshold')
ylabel('counts of synapse')
```

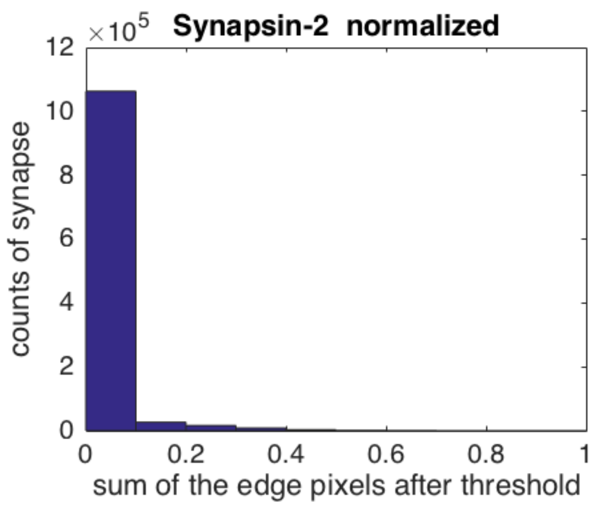
We try to look at the actual synapsins channel image to get rid of the mislabeled synapse.
 It was not obvious as to how we can define a local bright spot on the image as qualified candidat
e or not.
 As a result we tried to accomplish this feat by first thresholding the image and then subjecting
to the neighborhood of each labeled synapse in the pivot file to two different metric.
 One based on the number of non-zero pixel in the neighborhod, and one based on the number of edge
 pixel in the neighbood.
 For the non-zero pixel metrix, it turned out that for most labeled synapse a value of 0 was obtai
ned. We will look into better ways of quantifying this problem, but for now we just keep all the s
ynapse position
 with values above a 0.




Synapsin-1 normalized
×10^5
counts of synapse
integrated area of brightness after threshold
0 2 4 6 8 10 12
0 0.2 0.4 0.6 0.8 1


Synapsin-1 normalized
×10^5
counts of synapse
sum of the edge pixels after threshold
0 2 4 6 8 10 12
0 0.2 0.4 0.6 0.8 1


Synapsin-2 normalized
×10^5
counts of synapse
integrated area of brightness after threshold
0 2 4 6 8 10 12
0 0.2 0.4 0.6 0.8 1


Synapsin-2 normalized
×10^5
counts of synapse
sum of the edge pixels after threshold
0 2 4 6 8 10 12
0 0.2 0.4 0.6 0.8 1