ECE 532: Full lab exercise

Pancreatic tumor segmentation and detection of tumor cells migration based on white light and second harmonic generation microscopic images

Md Abdul Kader Sagar Adib Keikhosravi

1. **Introduction:**

The objective of this lab is to automatically find a specific type of collagen alignment around pancreatic tumor ducts. Since collagen has been shown to play a significant role in cancer metastasis, understanding of its behavior around tumor ducts has brought researchers attention recently. In this project we are looking for “tear drop” shaped structures around pancreatic tumor ducts, which might play a significant role in metastasis of this type of cancer.

1. **Overview:**

Pancreatic cancer is the fourth leading cause of worldwide cancer-related mortality, with an overall 5-year survival rate of 1–5% [1]. Collagen, the most abundant protein in vertebrates, forms the structural network of the extracellular matrix (ECM) in tissues and can vary in structure depending on type. Collagen alignment around tumor cells is proposed as potential biomarker for cancer metastasis [2]. Recently at our group, LOCI (Laboratory for Optical and Computational Instrumentation), researchers have found a special collagen alignment around pancreatic tumor ducts, which are called “tear drop” shape collagen alignment. First studies show a relationship between these shapes and tumor cell migration. In fact tumor cells push collagen fibers to move and migrate to other parts of the tissue, which they cause a stretched shape of collagen. Figure 1 shows two samples of these shapes. Two upper images show the registered SHG and fluorescent images and green circles show where the tear drop shapes happen in these images. Two lower images show the corresponding white light images for upper images. In this project we only use SHG images, which purely show collagen structure of tissue, and white light images and we don’t need fluorescent images at this point.



Figure 1 two samples of tear drop shape in pancreatic cancer tissue in SHG and white light image

There are two main steps in this project: first, is to segment tumor ducts from white light images, and second step is to classify collagen alignment around previously detected tumor ducts as tear drop shape or not. For the first step we use the information from both SHG and white light image to find the tumor ducts. For second step we use features related to fibers alignment around tumors to find tear drop structures. For the first step we use Hough Transform and for the second step we will use neural network classification.

**Theory of the Hough Transform**

In automated analysis of digital images, a subproblem often arises of detecting simple shapes, such as straight lines,circles or ellipses. In many cases an edge detector can be used as a pre-processing stage to obtain image points or image pixels that are on the desired curve in the image space. Due to imperfections in either the image data or the edge detector, however, there may be missing points or pixels on the desired curves as well as spatial deviations between the ideal line/circle/ellipse and the noisy edge points as they are obtained from the edge detector. For these reasons, it is often non-trivial to group the extracted edge features to an appropriate set of lines, circles or ellipses. The purpose of the Hough transform is to address this problem by making it possible to perform groupings of edge points into object candidates by performing an explicit voting procedure over a set of parameterized image objects.

The simplest case of Hough transform is the linear transform for detecting straight lines. In the image space, the straight line can be described as *y = mx + b* and can be graphically plotted for each pair of image points *(x, y).* In the Hough transform, a main idea is to consider the characteristics of the straight line not as image points , , etc., but instead, in terms of its parameters, i.e., the slope parameter *m* and the intercept parameter *b*. Based on that fact, the straight line *y = mx + b* can be represented as a point *(b, m)* in the parameter space. However, one faces the problem that vertical lines give rise to unbounded values of the parameters *m* and *b*. For computational reasons, it is therefore better to use a different pair of parameters, denoted and , for the lines in the Hough transform. These are the Polar Coordinates. The parameter represents the distance between the line and the origin, while is the angle of the vector from the origin to this closest point. Using this parameterization, the equation of the line can be written as

Which can be rearranged to . In this lab we will use Hough transform as a method for finding the information about collagen fiber direction. Because all the points on one line in Cartesian space will turn into sinusoids in Hough space meeting at one point.

**Neural network**

Neural network denotes a set of interconnected neurons, that influence each other by their computation. Analogous with the biological neural networks, the output of one neuron is the input of other neurons. For better understanding let’s figuratively divide the neural network into three parts:

• *architecture* – represents a structure of the network, how its neurons are connected. It can be simply imagined as a view on the network from outside.

• *active phase* – is an opposite to the architecture. It describes the inwards of the network– what happens from the moment, when the input enters the network till the computations reach its output.

• *adaptation* – is a networks reaction on the ongoing computations. It denotes alterations of the neurons weights.

Let’s shortly summarize the whole function of the neural network. At first we structure the neurons to shape some architecture. We have certain idea, what the network should do, which task should it fulfil. Consequently, we prepare a set of learning data that fits in the idea. Now we are ready to start the learning process – adaptation. When the adaptation is finished, we can start to use the network by inputting the data and letting them compute to start the active phase.

1. **Warm up**

**3.1. Creating binary mask from SHG and white light image**

A very basic image processing step is to exclude pixels that have intensity values larger or smaller than a specific value. You can create a mask based on the threshold value and use the mask to exclude unwanted pixel. For example the following code removes all the values that have intensity lower than 0.6 and creates a mask. The mask is used on the original image to create a modified image.

**Code**

a=imread('cameraman.tif');

[m n]=size(a);

bw=im2bw(a,0.6);

imshow(bw)

b=a.\*uint8(bw);

imshow(b)

Using this line in the above mentioned code will discard the pixel larger than 0.6.

*[NOTE: 0.6 is the normalized value, it is actually 0.6\*255=153]*

**Exercise:**

Can you write a code to keep the values between 100 and 200 in an RGB image discard the rest of it?

Erosion and dilation are two morphological operation in image processing. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. The size and shape of the structuring element will define how the output will look like. The element can be diamond, disk and octagon.

To be more specific, for dilation, the value of the output pixel is the maximum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to the value 1, the output pixel is set to 1.

And for erosion, the value of the output pixel is the minimum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to 0, the output pixel is set to 0.

For creating the structuring element MATLAB has a built in function. strel

For example if you run the code

a = imread('rice.png');

bw=im2bw(a);

imsow(bw)

se = strel('disk',1);

erode = imerode(bw,se)

figure, imshow(erode)

In this code it can easily noticed that the small noises are gone. To restore the mask in the original size use dilation with the same structuring element.

erode = imdilate(erode,se);

**Exercise:**

1) Can you write a code to find the edges off all the rice?

2) Write a code to find all the mask containing rice of the area between 200 to 300? [hint use ‘bwareaopen’]

3) Create a convex hull of the mask found in question 2.

**3.2. Training and testing neural network classifier**

Data for classification problems are set up for a neural network by organizing the data into two matrices, the input matrix X and the target matrix T.

[x,t] = wine\_dataset;

The next step is to create a neural network that will learn to classify the wines. Since the neural network starts with random initial weights, the results of this example will differ slightly every time it is run. The random seed is set to avoid this randomness. However this is not necessary for your own applications.

setdemorandstream(391418381)

Two-layer (i.e. one-hidden-layer) feed forward neural networks can learn any input-output relationship given enough neurons in the hidden layer. Layers which are not output layers are called hidden layers.

net = patternnet(10);

view(net)

The NN Training Tool shows the network being trained and the algorithms used to train it. It also displays the training state during training and the criteria which stopped training will be highlighted in green. Now the network is ready to be trained. The samples are automatically divided into training, validation and test sets. The training set is used to teach the network. Training continues as long as the network continues improving on the validation set. The test set provides a completely independent measure of network accuracy.

[net,tr] = train(net,x,t);

nntraintool

plotperform(tr)

The mean squared error of the trained neural network can now be measured with respect to the testing samples. This will give us a sense of how well the network will do when applied to data from the real world. The network outputs will be in the range 0 to 1, so we can use vec2ind function to get the class indices as the position of the highest element in each output vector.

testX = x(:,tr.testInd);

testT = t(:,tr.testInd);

testY = net(testX);

testIndices = vec2ind(testY)

**4. Lab exercise**

**4.1 Fiber structure feature detection and classification**

H&E and SHG are two modalities of microscopy as described earlier. For identifying the teardrop structures, we need to combination of SHG and H&E images. This will require some image processing to properly mask out the cancerous region. The H&E and SHG images are provided in their respective folders. We are giving some examples of cancerous region and how they look like in both H&E and SHG images

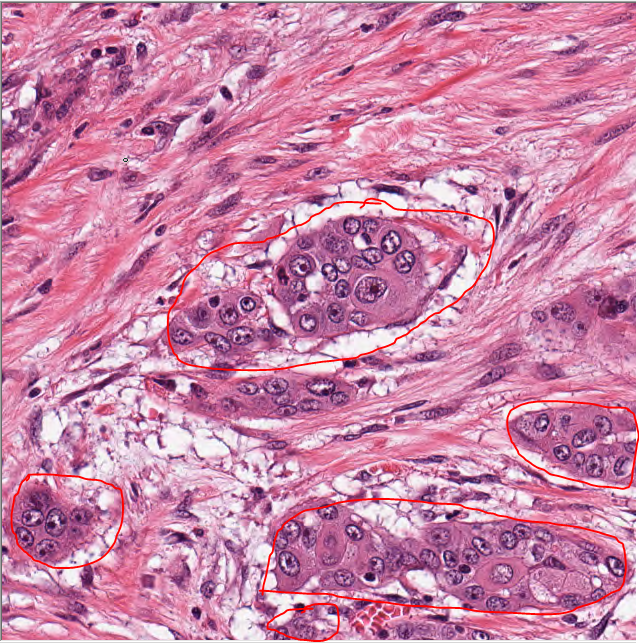


Figure 2 The cancerous region are marked as by red color.

C:\Users\sagarLOCI\Google Drive\LOCI\pattern recognition project\For AdibSagar\tumor 2000 #145 G3 SHG_C1.ome.tiff

Figure 3 SHG image for the same segment as figure 2

You might notice from the SHG image the absence of collagen in the cancerous region. And the collagen fibers converge to the both side of the duct (cancerous region) to form a teardrop structure.

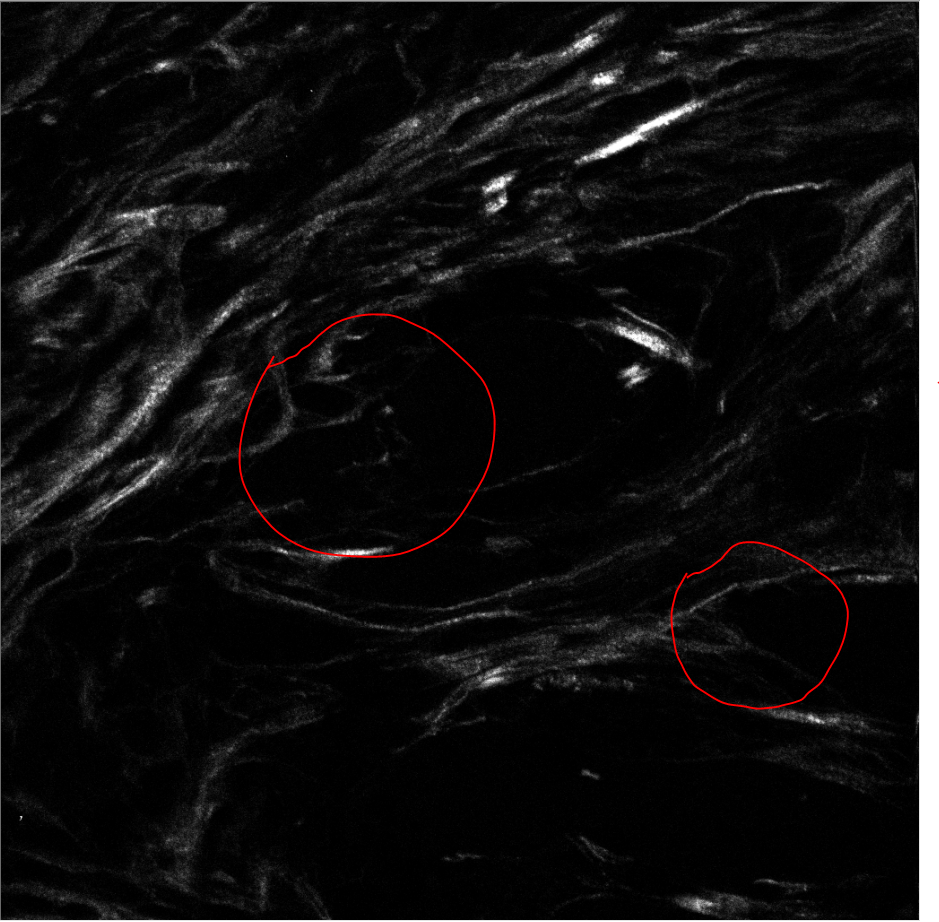


Figure 4 The tear drop structure regions are marked by red color.

For this step of lab you should use both the SHG image and H&E image to segment out the cancerous reigns. Use the property that cancerous region do not contain collagen fibers in the SHG image. For clarity we are giving some more examples of cancerous region:

|  |  |
| --- | --- |
|  |  |

Figure 5 Two more cancerous images

For creating the mask you can use this piece of code:

%%%%main code started

a\_mas=imread(SHG\_INPUT\_IMAGE);

a\_HE=imread(HE\_INPUT\_IMAGE);

[m n f]=size(a\_HE);

%

a\_mas=imresize(a\_mas,[m n]);

a\_gray=rgb2gray(a\_HE);

a\_mas=im2double(a\_mas);

a\_SHG=a\_mas;

bw=im2bw(a\_mas,0.04);

bw=bwareaopen(bw,size\_bwopen);

imshow(bw)

%%%creating intensity mask to remove white

a\_gray=rgb2gray(a\_HE);

temp=zeros([m n]);

for i=1:m

for j=1:n

if(a\_gray(i,j)<190)

temp(i,j)=a\_gray(i,j);

end

end

end

temp= mat2gray(temp);

figure

imshow(temp)

figure

imshow(bw)

se = strel('disk',15);

mask\_SHG = imdilate(bw,se);

figure, imshow(mask\_SHG);

%%%%%%%%%%%%%%mask creations

mask\_HE=im2bw(temp);

m=~mask\_HE.\*~mask\_SHG;

mask2=m+mask\_SHG;

mask2 = ~(bwareaopen(~mask2, 1000));

rgbImage=a\_HE;

mask2 = bwconvhull(~mask2,'objects');

mask2 = bwconvhull(mask2,'objects');

% bw1=mask\_HE&mask\_SHG;

figure, imshow(mask2);

mask=uint8(~mask2);

redPlane = rgbImage(:, :, 1);

greenPlane = rgbImage(:, :, 2);

bluePlane = rgbImage(:, :, 3);

% Do the masking.

maskedRed = redPlane .\* mask;

maskedGreen = greenPlane .\* mask;

maskedBlue = bluePlane .\* mask;

% Combine back into a masked RGB image.

maskedRgbImage = cat(3, maskedRed, maskedGreen, maskedBlue);

figure, imshow(maskedRgbImage)

bw1=bwareaopen(bw,200);

figure

imshow(bw1+mask2)

**4.2 Feature extraction using Hough Transform**

[H, theta, rho] = hough(BW) computes the Standard Hough Transform (SHT) of the binary image BW. Use the hough function to detect lines in an image. The function returns H, the Hough transform matrix. theta (in degrees) and rho are the arrays of rho and theta values over which hough generates the Hough transform matrix. BW can be logical or numeric, and it must be real, 2-D, and nonsparse.

Use the sub-images created in previous section to calculate Hough Transform of each one. The calculate three important features which are respectively the maximum value of the Hough matrix, average of all the elements in Hough matrix, and average of nonzero elements of Hough matrix.

*Note: since creating a feature set as training data needs so many images and tuning the masking parameters for each of them, we have provided two .mat files which contains the feature set, which is provided as ‘A.mat’ and desired output, which is provided as ‘b.mat’. You can use these features in next step to train and test the classifer.*

**4.3 Classification based on neural network**

As the last step you need to classify the provided feature set using neural network. Neural network is one the most powerful classifiers than can fit any curve which is going to be the separator of the classes. Use the instructions provided in section 3.2. to classify the data. As a hint try different numbers of neurons for the hidden layer to find the best accuracy. You should be able to classify the data with almost 90% accuracy.

To show the performance of your network after testing you can use this piece of code:

plotconfusion(testT,testY)

[c,cm] = confusion(testT,testY)

fprintf('Percentage Correct Classification : %f%%\n', 100\*(1-c));

fprintf('Percentage Incorrect Classification : %f%%\n', 100\*c);

plotroc(testT,testY)

**References**

[1] Siegel, Rebecca, Elizabeth Ward, Otis Brawley, and Ahmedin Jemal. "Cancer statistics, 2011." *CA: a cancer journal for clinicians* 61, no. 4 (2011): 212-236.

[2] Campagnola, Paul. "Second harmonic generation imaging microscopy: applications to diseases diagnostics." *Analytical chemistry* 83, no. 9 (2011): 3224-3231.