3) Image segmentation

Import a picture of someone's hand using

imread('hands2.jpg');

In class we talked about segmenting grayscale images but not color images. You should try the remainder of this exercise using the following set of different images:

A: The "red" channel from the color image (this is the first index in the third dimension

B: The "green" channel

C: The "blue" channel

D: The result of rgb2gray

E,F,G: The three channels that are the result of the rgb2hsv, a mapping that converts (red, green, blue) to (hue, saturation, value)

- Display the image
- Using Otsu's method, attempt to segment the image into hand/background
- Using some edge detectors (canny, sobel, prewitt, etc) see if you can segment the hand from the background. Remember, some of the edge detectors have additional parameters that you can adjust
- What is the area of the hand?
- Can you find some parameters or choice of RGB mapping that allows you to find the area of the fingernails?
- Once you have chosen parameters, see if they work for 'hands1.jpg' as well

4) Convolution theorem

We didn't talk about it in class, but there is a deep connection between Fourier transforms and convolutions called the "convolution theorem". It says that

$$F(x \otimes y) = F(x) \cdot F(y)$$

in other words, the Fourier transform of a convolution is the same as the element-wise product of the Fourier transforms of the components. While this has many applications, we're going to look at how one can use either the "naïve" or "fft" way to calculate convolutions and see which is faster.

- Set up 10 square kernels ($a_1, a_2, a_3, \ldots, a_{10}$) of random numbers with sizes that range from 3-300 and one test set of data (b) which is magic(600);
- Calculate the convolution of the b with each of a_i using the conv2 function. In addition to saving the result of these convolutions, use the tic and toc commands to measure how long the computation takes. Make sure to run it a few times and average the results, some timing issues need initialization and/or your computer running youtube or whatever.
- Calculate the convolution of the b with each of a_i using the convnfft function (on the wiki or on the mathworks fileExchange). In addition to saving the result of these convolutions, use the tic and toc commands to measure how long the computation takes
- Confirm or deny if the two algorithms give the same results using the "isequal" function. A similar function that supports nans is "isequalwithequalnans".
- What is the largest relative difference between the two functions? Calculate the maximum value of (method1-method2)/method1. Would you be comfortable using one method instead of the other?
- Plot the log(executionTime) vs log(kernelSize) for both functions. When would you want to use one method instead of the other?