**MATLAB For Scientists**

**HW7**

**Instructions**

* Complete each question below in an **individual** script or function. Each individual script should be named according to: HW#Q#\_script.m, or HW#Q#\_fun.m where <#> indicates the corresponding HW number or question. These are the scripts that will be considered for grading.
* Submit all files (**including** the provided data, but **excluding** any tables/files that are saved as a result of the HW question) as a compressed .zip folder with the name **<HW#\_YourCodeName.zip>.**
* Each script **must** run entirely through without error. If you did not finish part of a question, comment it out so it runs.
* Suppress all intermediate outputs other than your answer to the question; only the answer should display on the command line.
* Absolutely no hard coding beyond the minimum specified.
* **All plotting should now conform to our guidelines of publication quality figures**
* **Everything must be commented. Uncommented codes get zero credit!**

**Problems**

**Question 1 (4 pt)**

Using HW7Q1\_nemo.jpg, and write a script to do the following:

1. Read in the image and visualize using imshow
2. Create a 1x3 subplot of the R, G, and B planes of the image in grayscale
3. Using only manual pixel ranges (e.g., no cropping functions), select a region within the image that most closely isolates nemo’s face, and plot
4. Convert the image to grayscale and report the mean and median of the pixel intensity

**Question 2 (4 pt)**

Using HW7Q2\_landscape.jpg, write a script to do the following:

1. Split the image vertically down the center (how to find the center without hard coding?) and create two new images that combined make up the whole. On a 3x2 subplot, use the first row to plot the original two sides of the image, on the second row plot the grayscale converted image, and on the last row plot the binarized image.
2. Create a matrix that is the G plane of the right half of the image, and use this to replace the G plane on the left. Plot the resulting figure.
3. Create a 3x1 subplot to show the outcome of performing the following functions
   * 1. Multiplying left and right image sides
     2. Adding left and right image sizes
     3. Subtracting left and right image sizes

**Question 3 (6 pt)**

A patient who was diagnosed with a glioblastoma tumor by brain contrast imaging received a follow up scan 8 weeks later. Using HW7Q3\_ glioblastoma\_0wk.jpg and HW7Q3\_glioblastoma\_8wk.jpg, write a script to do the following:

1. Doctors would like to compare the tumors, but the images are different sizes. Use the resize function to create a new variable, im8\_scale that scales the image from week 8 to match the image size from the initial diagnosis at week 0. These images are located in glioblastoma. Report the size of the scaled image.
2. Convert the images that are now comparable to grayscale and plot on each in a 1x2 subplot.
3. Using the images defined in (b), find the indices of a pixel that is close to the center of the tumor using any method in each of the images. Make a new subplot that places a scatter marker on each image in blue that marks this location.
4. Using the pixels from (c), plot two lines on the same figure, where each line corresponds to the horizontal vector of pixel intensity that crosses through the index. Plot each in a different color, and include a legend + other formatting specifics. Can you detect a difference?
5. Write a for loop that goes from -10 to +10 from the index you chose, and creates a matrix that extracts the horizontal vector at each index of the loop. Using one figure, plot all vectors from week 0 in black and week 8 in red.
6. Do the same thing as in (e), but now with the complement of the image. Does this make a difference?

**Question 4 (2 pt)**

Cells grown on a nitrocellulose plate and stained the tetrazolium salts fluoresce with intensity that is proportional to their redox state. The image in ‘HW7Q4\_arrayplate.tif’ is a representative example of this. Import this file and convert to grayscale. The nitrocellulose paper is clearly visible. Use drawrectangle to obtain the coordinates for the paper vertices, and generate a mask that will filter anything outside of the paper edges. Report imshowpair to visualize the mask on the image.

**Question 5 (8 pt)**

High pixel intensity corresponds to areas within the image of high bacterial growth. Clearly, however, pixel intensity data is noisy. To identify trends in noisy data, moving averages are often performed in complex algorithms to smooth curves, which reduces noise and amplifies signal. The function CalculateSlidingAvg.m has been provided for you, which takes the following syntax: svec = CalculateSlidingAvg(vec,n), where vec is the vector you are looking to smooth, n is the width of the moving average window, and svec is the output which is a smoothed vector. Use the grayscale still frame called HW7Q5\_gray.tif to write a script that does the following:

1. Defines a vector vecX that corresponds to the pixel intensity across the horizontal midpoint of the image and plot it as a function of distance (e.g., the x coordinates where x = 0 corresponds to the left-most pixel).
2. Smooths vecX by looping through sliding windows of 2, and up to 65, at increments of 5, and make a new figure that contains all smoothed vectors as a function of distance, including the original vector.
3. Plots the vector that you decide (from manually inspecting the results of (b)) best smooths the data such that the regions of the chamber are most clearly defined on a new figure.
4. Marks the two boundaries of the growth chamber by placing markers at each index on the figure generated in (c). Hint: This can be done in many ways that we have learned, such as diff(x).

**Question 6 (8 pt)**

Using the RGB image called HW7Q6\_rgb.tif, write a script to do the following:

1. Visualize the image using imshow

Ideally, intensity outside the growth chamber should be 0, since signal here represents fluorescence that is noise or coming from a different chamber, and not from the cells themselves. Clearly, in this image, background is too high. Removing this is called background subtraction.

1. Background subtracts by (1) converting the image to grayscale, (2) calculating the background using a representative region on the image and subtract this value from the entire grayscale image, and (3) contrast-adjusting the image with one of the functions we learned.
2. Plots a 1x3 subplot with the original image on the left, the background-subtracted image in the center, and the grayscale contrast-adjusted image on the right. Hint: Convert the background into uint8 to keep units consistent.
3. Generates and plots a mask for the chamber in the image using the equation of a circle: (x - centerX).^2 + (y - centerY).^2 = r^2. Assume centerX = centerY = 512, and r = 400, where x and y are pixel indices (how would you determine the center and radius without hard-coding?). The mask of the image should consist of the value 1 for any pixel that is located within the circle (less than r2), and 0 for any pixel located outside of the circle (greater than r2) equals 0. To plot the mask, it must be converted to uint8.
4. Plots a new figure of the contrast-adjusted image that is filtered by this mask.
5. Calculates and reports the average intensity of the background-subtracted image inside of the masked region in a complete sentence.