# BMI/CS 567 Medical Image Analysis
# University of Wisconsin-Madison
# Assignment #1

Instructor: Jeanette Mumford
**Due** Feb 7, 2018 *before* class begins

Please turn in the pdf or html version of a MATLAB live script that contains your code, output and text answers (when needed). Keep your code neat and limit output to what is specifically asked for. Name it `hwk1_yourname.pdf` and submit via canvas. Late homework will not be graded.

**(1)** Reading in a ppm file and differentiating between image display options in MATLAB.

(a) (3 points) Use Google or the MATLAB help to read in the `img_001.ppm` image that was distributed with this assignment. You *cannot* use a downloaded MATLAB function to do this, but must use a built in MATLAB function. After reading in the image, display the dimension of the 3D array and describe what each dimension (rows, columns, pages) represents. What are the smallest and largest intensity values in the array? How many pixels are there in the image?

(b) (1 points) Display the first 5 row values of the first column in the first page of the 3D array you read into MATLAB in part (a). Divide these values by 3 and display the output. What is wrong with the values and what caused this problem? Change the image to double precision and repeat. Did this fix the issue? Why or why not? Please make it a habit to *always* change an image to double precision when you read it into MATLAB.

(c) (2 points) Use Google or the MATLAB help to learn about the differences between image and imagesc. Using the *double* precision version of the image from part (a), make 2 separate figures (not a single, multipanel plot), where each one displays the green color channel of the image using image and imagesc. Display the images in grayscale. Do they look different? If so, why?

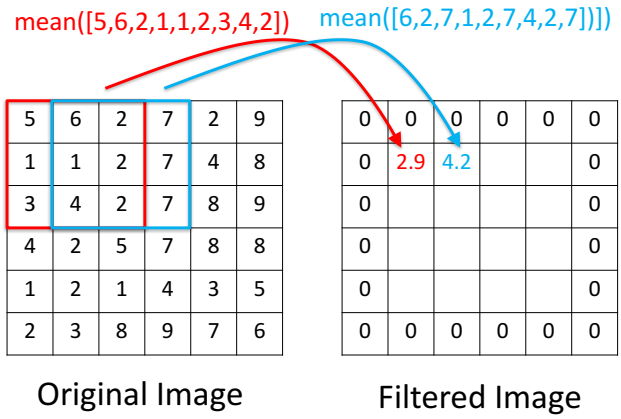
**(2)** Filtering an image. We will learn numerous ways to filter images throughout the semester. To grow your skills working with matrices in MATLAB, you will mean filter the image from the previous problem. Specifically, use the double precision version of the green channel. For this problem you will use a `for` loop and the `mean` function, but may not use any other filtering-specific MATLAB functions.

(a) (4 points) The image below describes how the mean filter works. The left hand side is the starting image and right hand side is the filtered image. For now, we'll ignore the edges and simply put 0s for the edges and fill in the rest. For each pixel construct a $3 \times 3$ neighborhood around it. The center of this neighborhood is the target in the new matrix (where your value will go) and the value is the mean of the 9 values in the neighborhood. The image illustrates the first two pixels. Filter the image and print out the values for the 100 pixels in the first 10

columns and rows. Also make a $1 \times 3$ panel plot where the left panel is the original image the middle panel is the filtered image and the right panel is the original minus the filtered. Label the images, accordingly. Subtracting images is often a nice way to see what has changed and also can be a helpful processing strategy for an image, for example to remove artifact.

(b) (5 points) Repeat what you did in the previous problem, but use a $10 \times 10$ neighborhood. Make sure you use exactly the right number of 0s along the edges. Hint: It isn't 1 layer of zeros as in part (b).

(c) (2 points) Looking at the differenced images from parts (a) and (b), what is one artifact that was in the original image that disappears after subtraction? What is one artifact that is introduced?




**(3)** (4 points) Changing the resolution of an image. You will again use the $605 \times 700$ green channel of the retina image for this problem and the goal is to change the resolution by resizing the image to be $121 \times 140$ (reducing by a factor of 5). The image below illustrates how this would be done for a factor of 3. Display the dimensions of the resized image and also display the original and resized image in a $1 \times 2$ panel plot, labeling the plots accordingly. You may not use any special MATLAB functions for this problem, other than a for loop and the mean function.