Image Processing in Analysis 1 Lab #1

What is Noise in an Image?

1. Consider the pairs of (noisy/clean) images below. Use these images to answer the following questions.
    (a) What makes an image "noisy"?
    (b) How do you tell that an image contains noise?


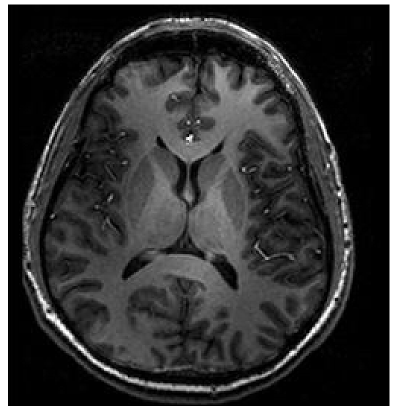

Figure 1: MRI of a human brain. Source: http://www.ohsu.edu/xd/research/centers-institutes/airc/ airc-7t-instrument.cfm


Figure 2: A boat. Source: http://dmmd.net/main_wp/research/denoising/

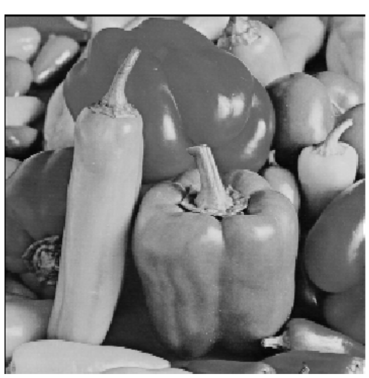
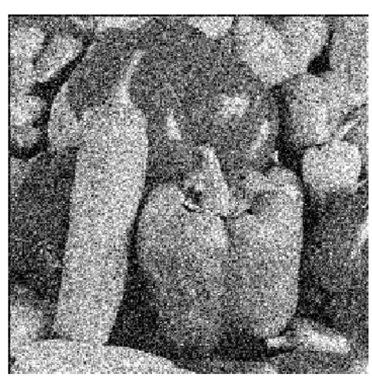

Figure 3: Peppers. Source: http://www.cs.nyu.edu/ ranzato/research/projects.html

Exploration: Modeling space of images as a metric space

2. Consider the following four-pixel (2×2) images obtained by entering the following code into Matlab or Octave. Notice that entering a semicolon at the end of a line suppresses the output. Try entering a line without the semicolon to see this.

```
M_1 = [5 10;0 5];
M_2 = [5 5;5 5];
M_3 = [5 0;10 5];
M_4 = [3 8;2 7];
M_5 = [0 10;0 10];
M_6 = [6 10;0 4];
subplot(2,3,1);imshow(M_1,[0 10]);title('Image 1');
subplot(2,3,2);imshow(M_2,[0 10]);title('Image 2');
subplot(2,3,3);imshow(M_3,[0 10]);title('Image 3');
subplot(2,3,4);imshow(M_4,[0 10]);title('Image 4');
subplot(2,3,5);imshow(M_5,[0 10]);title('Image 5');
subplot(2,3,6);imshow(M_6,[0 10]);title('Image 6');
```

   (a) How would you decide which are similar to each other? Think about metrics when answering this question.
   (b) Compare each image to Image 1.
   (c) Which is/are most similar?
   (d) Which is/are most dissimilar?

3. Now let's try some slightly larger images (5×5 images). Paste the following code into Octave or Matlab:

```
A = [12 62 93 0 22;
        12 62 93 0 22;
        12 62 93 0 22;
        12 62 93 0 22;
        12 62 93 0 22];
B = [0 0 0 0 0;
        20 20 20 20 20;
        90 90 90 90 90;
        20 20 20 20 20;
        0 0 0 0 0];
C = [12 62 93 0 22;
        0 45 30 29 0;
        12 62 93 0 22;
        80 10 30 63 48;
        12 62 93 0 22];
D = [12 62 93 0 22;
        20 20 20 20 20;
        12 62 93 0 22;
        20 20 20 20 20;
        12 62 93 0 22];
subplot(2,2,1); imshow(A); title('Image A');
subplot(2,2,2); imshow(B); title('Image B');
subplot(2,2,3); imshow(C); title('Image C');
subplot(2,2,4); imshow(D); title('Image D');
```

(a) Is Image C more similar to Image A or Image B?

(b) Is Image D more similar to Image A or Image B?

(c) What criteria are you using to determine the answer to these?

4. Enter the following lines of code one at a time, and examine what each one does. You should be able to figure out what each of the following operations does (with matrix inputs):

`-` `*` `.*` `^` `.^` `abs()` `sum()`

```
I_1
I_2
I_1-I_2
abs(I_1-I_2)
abs(I_1-I_2)*abs(I_1-I_2)
abs(I_1-I_2).*abs(I_1-I_2)        %How is '.*' different from '*'?
abs(I_1-I_2)^2
abs(I_1-I_2).^2                   %How does '.^2' differ from '^2'?
sum(abs(I_1-I_2).^2)
sum(sum(abs(I_1-I_2).^2))
sum(sum(abs(I_1-I_2).^2))^(1/2)
```

(a) The last line above computes the Euclidean distance between the image matrices I_1 and I_2. Compute the Euclidean distances between Image 1 and the other 5 images. Based on this metric, which is closest to Image 1? Which is farthest away?

(b) Now modify the Matlab code to compute instead the Taxicab distance between Image 1 and the other 5 images. Which image is closes to Image 1 based on the Taxicab metric? Is this different than your answer using the Euclidean metric?

(c) Modify the Matlab code once again to compute the *Sup norm distance* between Image 1 and the other 5 images. The sup norm distance between two matrices A and B is defined to be $d_\infty(A, B) := ||A - B||_\infty = \max_{i,j} |a_{ij} - b_{ij}|$. Compare your results to the first two metrics. (Note that rather than using a summation, here you will need a maximum!)

(d) Recall that you answered by eye whether Image C is more similar to Image A or Image B and whether Image D is more similar to Image A or Image B. Now that we have mathematically defined three different ways to measure the distance between two images, what can you say about these questions? Do your answers depend on which of the three metrics you use? Is this always the case?

Add your own noise to an image.

Paste this code into Octave or Matlab. What does it do?

```
A = [12 62 93 0 22;
        12 62 93 0 22;
        12 62 93 0 22;
        12 62 93 0 22;
        12 62 93 0 22;
        12 62 93 0 22;
        12 62 93 0 22;
        12 62 93 0 22;
        12 62 93 0 22;
        12 62 93 0 22];
[m,n] = size(A);
N= randi(60,m,n);
A_noisy = A + N;
subplot(1,3,1); imshow(A);
subplot(1,3,2); imshow(A_noisy);
subplot(1,3,3); imshow(N);
```

A couple take-home messages:

A Image processing problems are generally more complicated than they appear at first glance: questions like "is there noise in this image?" or "how do we know when we are finished denoising?" are often somewhat subjective.

B Your eyes are *really amazing* image processors!

C Computing (using analysis techniques and the mathematical modeling process) can often help automate image processing functions... we often determine whether a method or metric is useful by how closely we can mimic our eyes' observations.

A preview of things to come

One exciting application of real analysis involves developing solutions to the image denoising problem! Suppose we have a mysterious, unspecified function that measures the noise in an image. We are looking around in the "space of possible images" and we want to find the direction we can move so that the noise decreases the most.