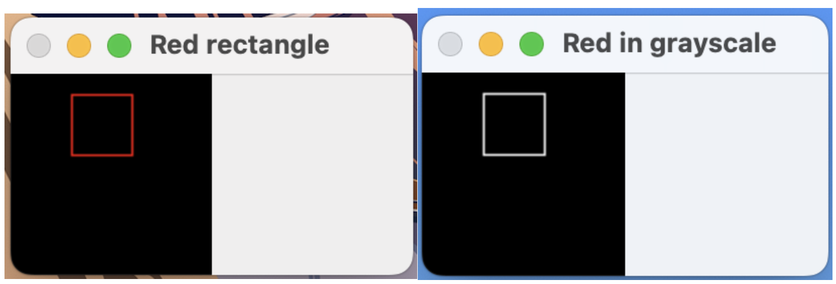
CS 282: Report on Programming Assignment 1
OpenCV Exercises: Image Enhancement

Submitted by:
Mark Jeremy G. Narag
2014-64423

1. **Computation:**

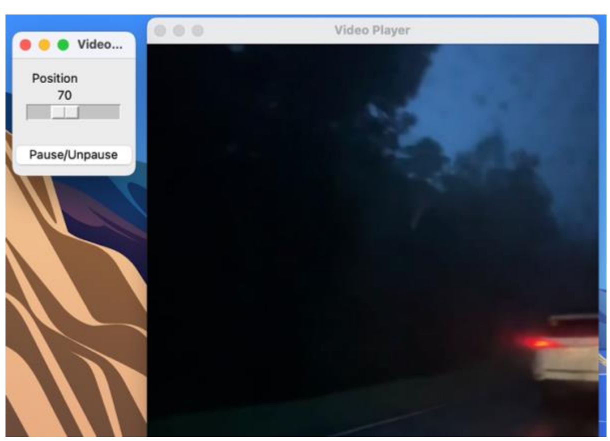
   For parts a to c: I was able to use openCV to solve for inverse of a matrix, eigenvalue of a matrix, and even solving for x in Ax = b given matrix A and b. The matrix given is actually challenging since each element of the cell in the matrix has three components. We can think of this as the RGB value of an image where each component represents the color channel.

   Parts d and e: drawing a red rectangle with corners at (30,10) and (60,40). Then we display it in grayscale:



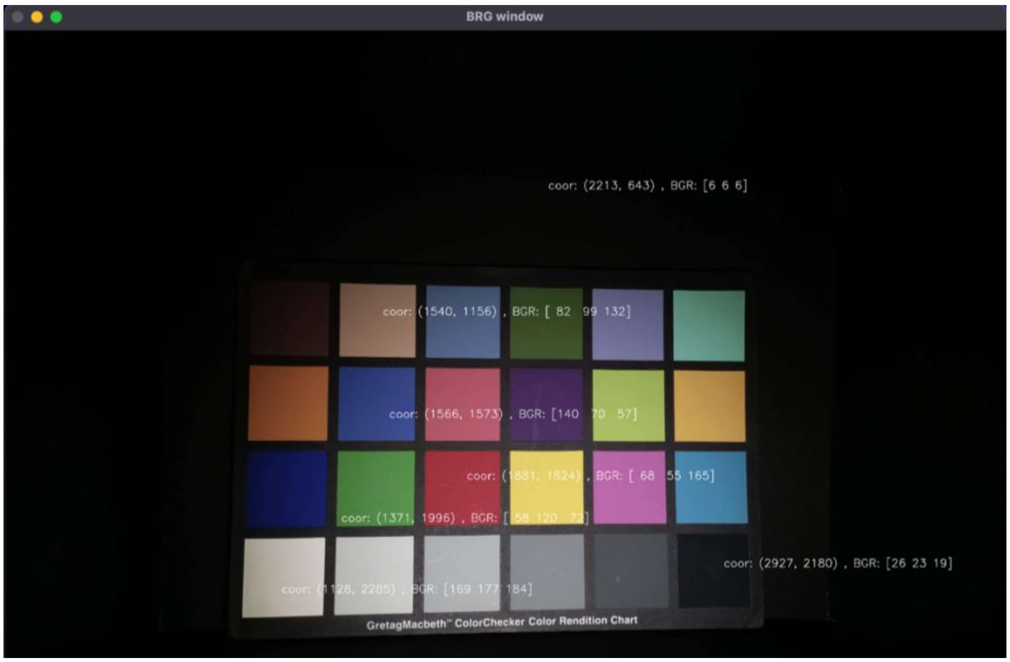
2. **Video player with controls:**

   Here, I was able to make a control button for playing the video (test.mov) I captured going to Masungi Georeserve. I integrated OpenCV and Tkinter to do this. Tkinter is a python library for creating graphical user interfaces (GUIs) which we will need for the controls. I was not sure regarding the "10 increments" in the problem so I just used 10 millisecond increments. I had a hard time doing this and I would like to thank the forums online that helped me do this.



3. **BGR values**

Here, I used my Macbeth chart illuminated by an LED. I chose this as an example so we can easily check if the BGR values are making sense. Shown below is the output of my code. It displays the coordinates (coor) of the image where I clicked it, as well as the BGR values. On the black background, we can see that the BGR values are low – [6,6,6] as expected. For the red patch in column 3, row 3, it gives the value [68,55,165] which make sense since the r channel have a large value. Same observation can be drawn to other color patches.



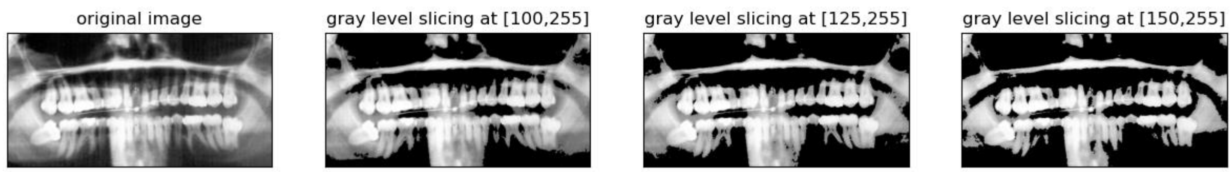
4. **Edge detection**

Shown below is a screenshot of a live canny edge detection method using the webcam of my camera. The first window is the original captured colored image. The second window is the grayscaled version, and the third window is the canny edge detection in which I used a lower and upper intensity threshold value of 50 and 125. We can easily change these values if we want. As we can see, our Canny Edge was able to detect the patterns in my shirt even the subtle contours of the 50 peso bill. However, it failed to detect my neck.

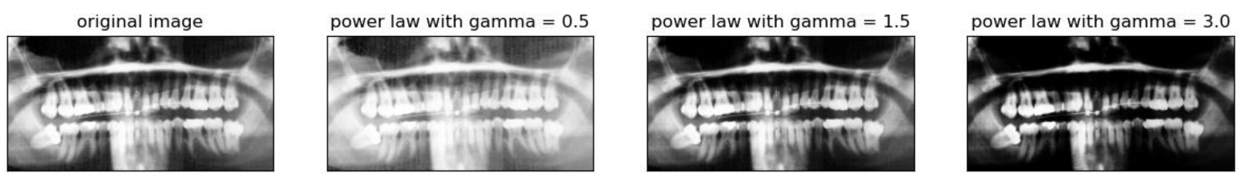
5. Image Enhancement

a. dental.gif

For an x-ray image, we can apply *gray-level slicing* to highlight only the important parts of the x-ray, which in this case, it the teeth. Shown below is the output of gray-level slicing using different threshold values of 100 to 125, 125 to 255, and 150 to 255. As we can see, as we increase the lower threshold value, some of the teeth were not detected anymore. But if we decrease the threshold value, we only see a slight difference on the original. Gray-level slicing may not be appropriate for this image.

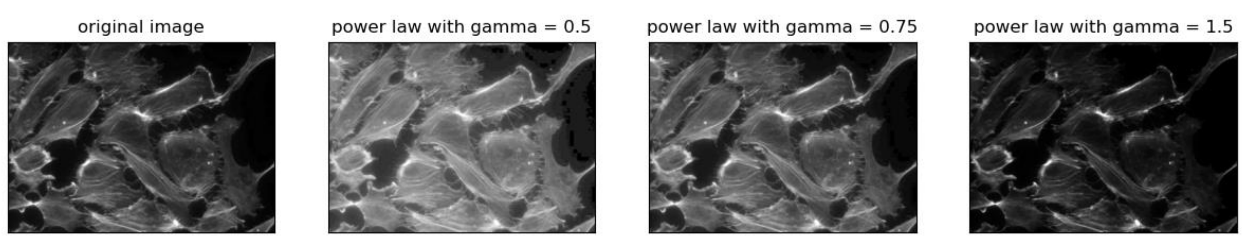


We can then try another method which is the *power law transformation*. With this transform, we can intensify the teeth area while lessening the intensity of the background. When gamma is less than 1, the whole image becomes brighter but when we increase the gramma greater than 1, we now observe that the original bright areas (the teeth) gets brighter, while the background gets darker. With gamma = 0.3, we were able to focus the teeth better compared to the original.

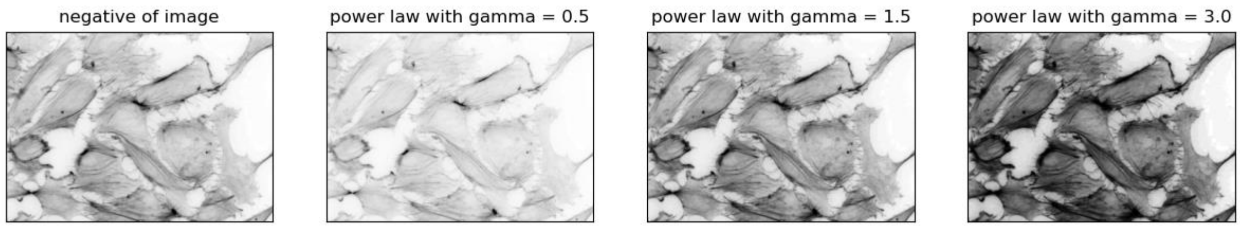


b. cells27.jpg

The original image of the cells is relatively dark. We can again apply a power law to enhance the image. In this case, we want to brighten the image so the gamma should be less than 1. As we can see from gamma = 0.5, the small threads are now apparent. With gamma greater than 1, as expected, the image gets darker.

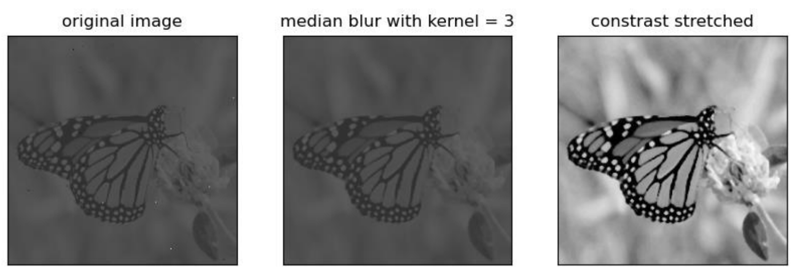


Another way is to actually get the negative of the image as shown below. And from that, we can again apply a power law. In this case, we want gamma greater than 1 to intensify the details of the cell. With gamma = 3.0, we can now see the details of the cells clearly.



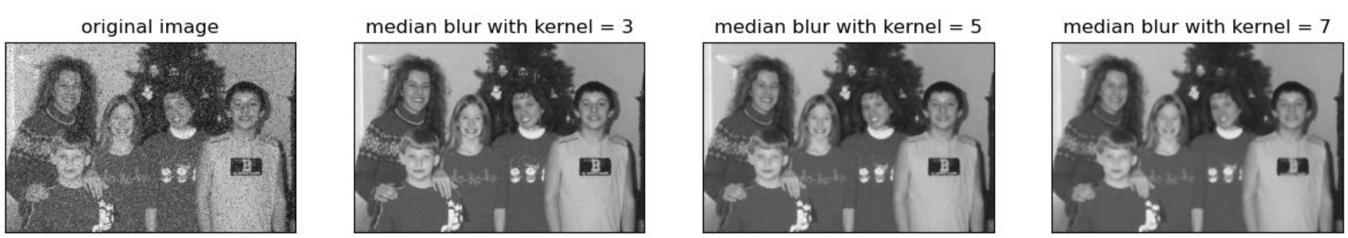
c. butterfly.gif

The original butterfly image have subtle salt and pepper in it so the first thing to do is to remove it using a median filter with kernel = 3. Bigger kernel size will blur the image. After "cleaning" it, the image is in low contrast so we can stretch the pixel values from 0 to 255 to make it visually appealing.



d. momandkids.jpg

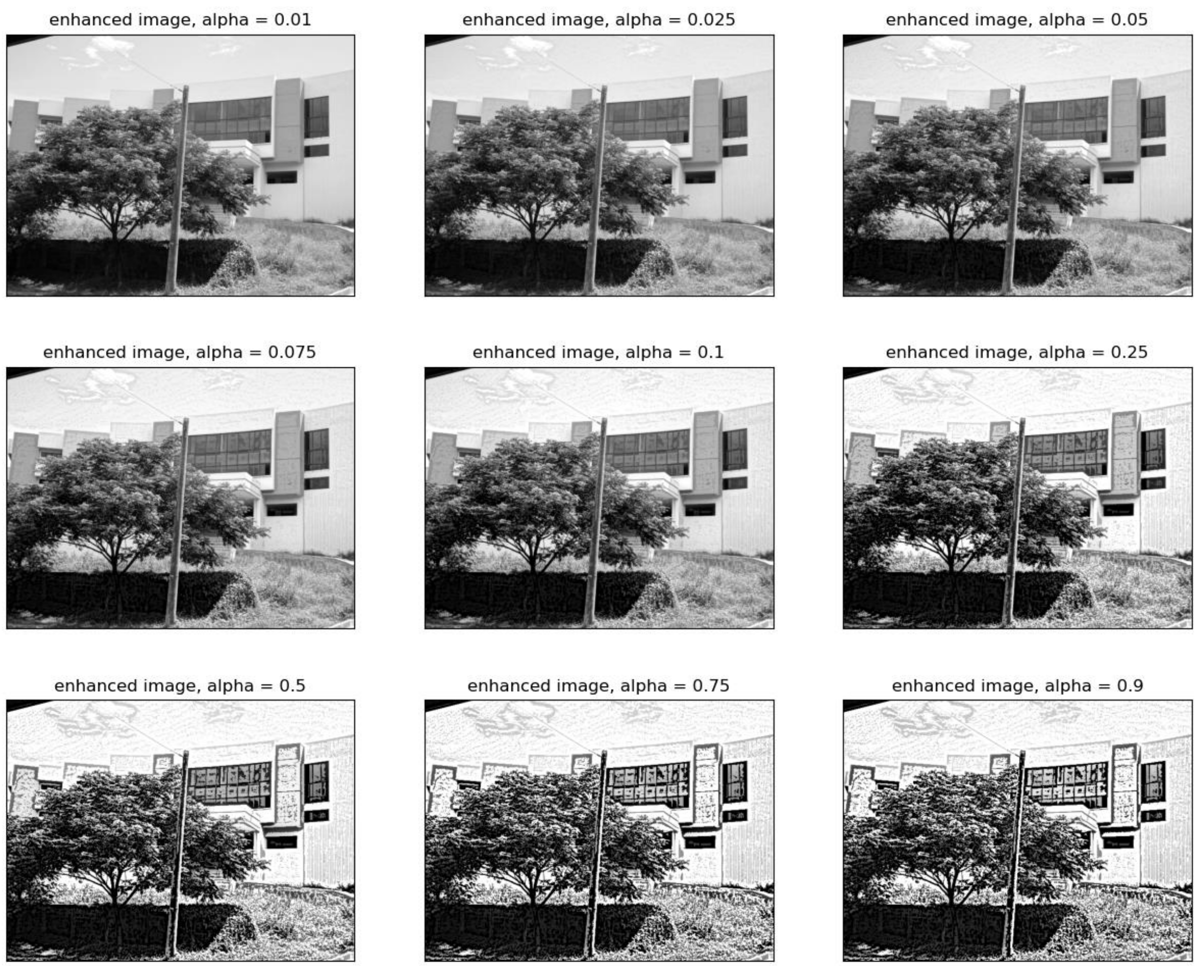
The mom and kids photo is full of salt and pepper noise. We can enhance this using a median filter to remove the noise. On the image below, I used different kernel size. As we can see, increasing the kernel size blurs the image. A kernel of size 3 by 3 is already enough for the job.



6. Unsharp Masking in the Spatial Domain

Shown below is the original image, the blurred image using gaussian blur with sigma = 3, and the mask. The mask is just the difference of the blur from the original image.

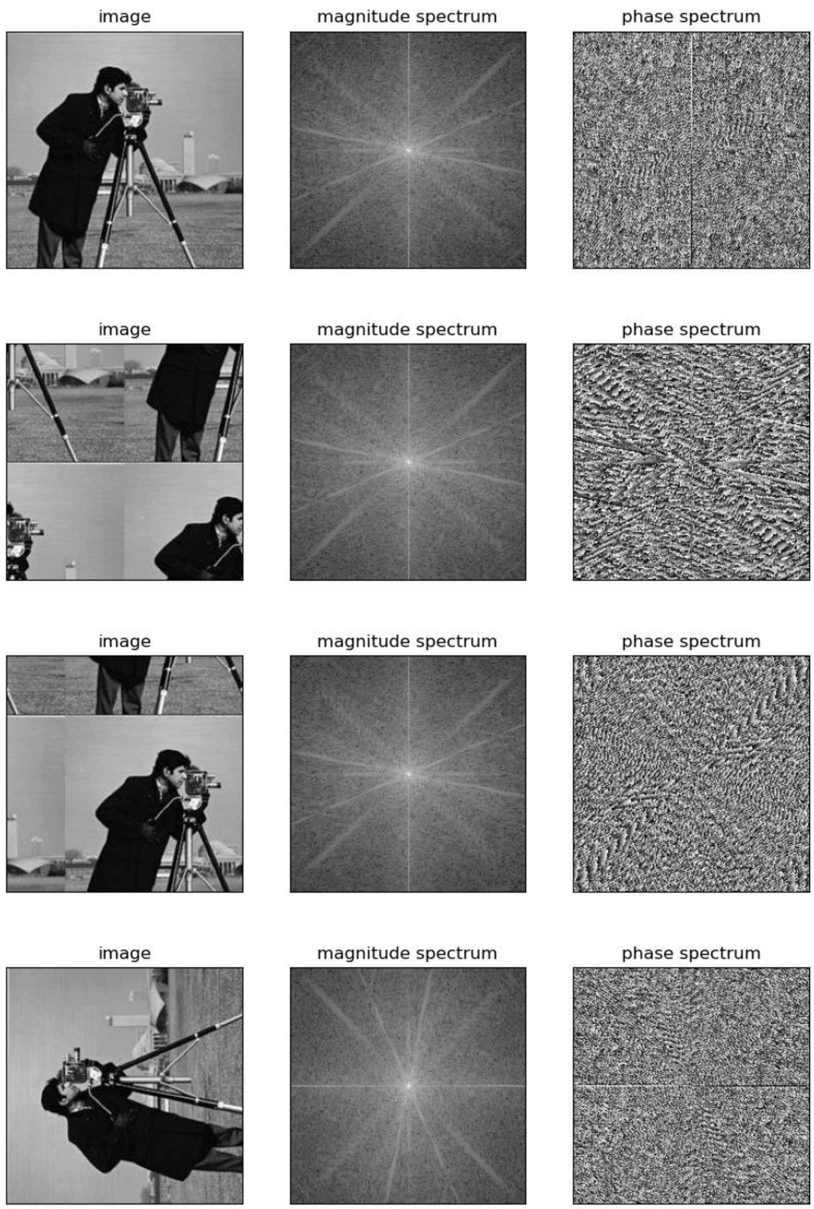
Now, to enhance the image, we can add the mask to the original image. We will control the fraction (gamma) of the mask to add on the original image. Below is the result for the different gamma:



Unsharp filtering enhances edges and details of an image but it can also introduce artifacts. We need to carefully choose the parameters such as the amount of blurring and the fraction added back. If these parameters are chosen appropriately, the enhancement of edges and details can be real and visually pleasing, with minimal artifacts as can be seen in lower alphas = 0.01, 0.025, 0.05. If we go higher than these alpha values, the artifacts can become more pronounced. We can see these in alpha > 0.1

7. Translation and Rotation Properties of the 2D Fourier Transform

Shown below is the resulting magnitude and phase spectrum of the FFT of a given image.



As we can see, the magnitude spectrum did not change in cameraman 2 and 3 even though the image seems to be jumbled. This is because the rearrangement of pixels does not alter the underlying spatial frequency content of the image. It simply changes the spatial arrangement of the data. The phase spectrum changes however.

In cameraman 4, since we rotated the image, the magnitude and phase spectrum also rotated with respect to cameraman1. Remember that in the frequency domain, the magnitude of the FFT represents the amplitude of different spatial frequencies. We can consider these frequencies as "gradients" or variations in pixel values in different directions (horizontal, vertical). Thus, when we rotate the image, the spatial frequencies change orientation because the original horizontal and vertical components shift in the rotated image. This change in orientation of the spatial frequencies results in a corresponding rotation of the magnitude spectrum.