Report of Assignment 5

CH22B007

Question 1

What to do

This question involved using the concept of matrix manipulation to get kernel operations done on an image.

Method of Solving

- The first thought process was to use **3 for loops** for the image, which will result in computations of the order of 10^6. This was clearly a very inefficient method, and required a lot of time, around 20 mins to compile and give an output.
- The better method involves thinking about the kernel operations as the linear combinations of shifted matrices. If the size of an image is [r c 3], then a matrix of size [r-5 c-5 3] was defined, for a 5*5 kernel.
- The kernel elements, (i,j) are multiplied and added and vary from row i to endrow r-5+i, and correspondingly column j to endcolumn c-5+j.
- The datatypes needed to be typecasted as double, and as uint8 at the end for correct results. Otherwise a number more than 255 is set at 255 and saturation increases, where it shouldn't be like.
- I ignored the boundaries of the original image matrices, so the final size has 4 rows and 4 columns less than the original image.
- Similar can be done if a 3*3 matrix is taken as well.
- k1, k2 and k3 are the three kernels whose o/p we can get. Similarly, a few commented matrices can also be used.

Plots

I am plotting 2 figures, in which the first contains a subplot of size 1*2, and it contains the original and final images, i.e. before and after the image manipulation.

The second one contains a histogram of previous and later cases.

Figure 1: Gaussian Blur

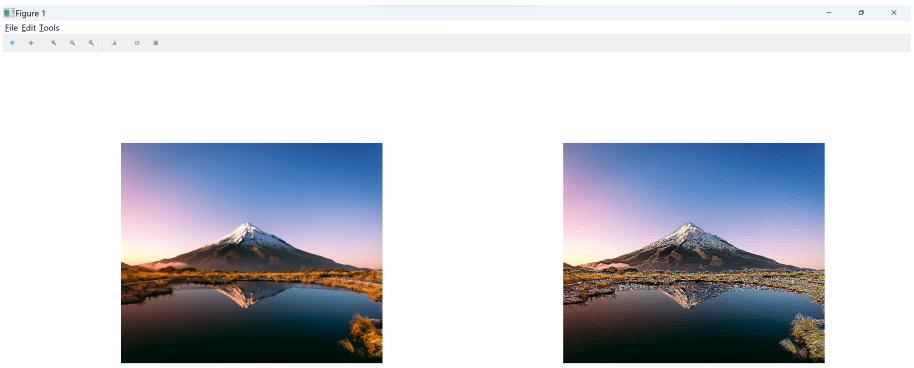

Figure 2: 'Original image vs Prewitt-convolution-kernel modified image'

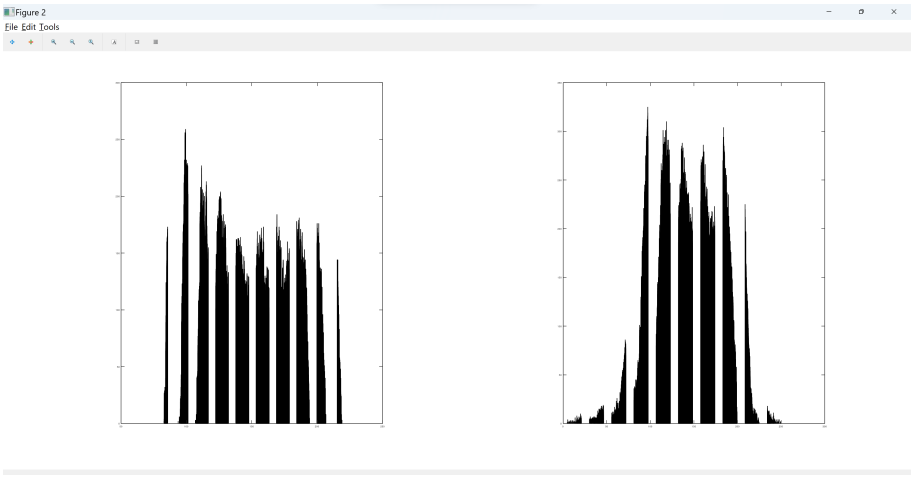

Figure 3: Old vs New histogram of the Prewitt kernel case

Question 2

What to do

The question asked us to plot a function containing at least 3 constants, and the add noise to it. We then have to plot a function which is the best fit to the noise-added version of the polynomial function.

Method of Solving

- I generated a random array from 0 to 1, of vnum values named xvalues.
- The xvalues array was sorted using the sort(xvalues) function.
- The noise was created by drawing back the array size by 0.5, (to make it go from -0.5-+0.5) and then multiplied by 0.02, which is delta.
- **Polyval** is a function which I used to generate corresponding y value.

- After that, Noise was added, and then the function **polyfit** was used to solve the question, by finding the best fit polynomial function. This method utilizes the method of minimization of squares, i.e. the method of least squares.
- I created 2 figures which are displayed as follows.
- The fitted curve and the original curve are very similar, and that is visibly seen in the way the yellow and blue lines are very close and similar to each other.

Plots

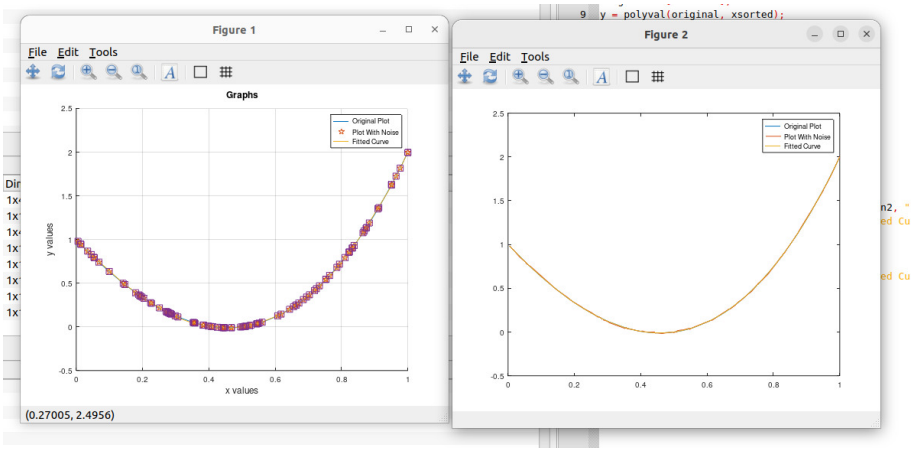

Figure 4: One with additional plotting, and other plain.

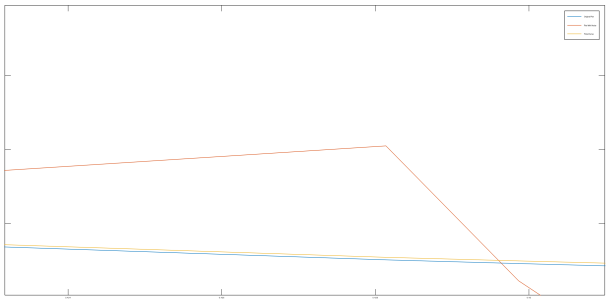

Figure 5: Zoomed in image of the 3 plots