Lab 1

Part 1

The first thing I notice about “Beautiful\_Green\_Picture.jpg” is that it is very green. I don’t think its obvious that it is light or dark (overexposed or underexposed respectively).

A pond with a fountain in the middle of it

Description automatically generated

When I split the picture into the respective RGB colours, the intensity of each colour becomes more apparent. Blue is clearly the darkest (least intense), followed by Red, then Green being the most intense (lightest).

A screenshot of a computer

Description automatically generated

This is just a rough guess though and we need to make a histogram of these separated RGB images to know for sure which one is the most intense and also look at each colour’s spread of intensity.

A screenshot of a computer

Description automatically generated

As we can see, Green is probably the most evenly spread but I think it is fair to say that all three colours point to the histogram being underexposed due to the lack of high intensity pixels. Red has the largest contrast.

Part 2

A screenshot of a computer

Description automatically generated

This histogram equalisation enhances detail and a difference can be seen in the pictures as well as the histograms. The full greyscale dynamic range is exploited more so in the equalised image than the original. Histogram equalisation is not always going to improve image quality as it often causes a change in mean brightness which is not necessarily a good thing.

A screenshot of a computer

Description automatically generated

Above, we see the optimised gamma corrected image. I made a for loop to calculate the “Perception Based Image Quality Evaluator” (PIQE) no reference image quality score.

According to the PIQE score, the image adjusted through Gamma correction is better, however, I think the one made using histogram equalisation is better due to there being more contrast.

A screenshot of a screenshot of a fruit store

Description automatically generated

Gaussian then Salt and pepper synthesisation.

A group of oranges in a store

Description automatically generated

Increasing sigma in the gaussian filter from 0.5 to 0.8 results in a much better result, particularly on the image where we introduced salt and pepper noise. The default gaussian filter did the job on the image with gaussian noise. We can see in both images that the noise introduced white spots onto the original images and these disappear once the filter is applied.

A screenshot of a computer

Description automatically generated

The figure above shows how the median filter tries to fix the images with salt and pepper noise in them. We learn that reducing the neighbourhood size from the default [3 3] leads to less noise being filtered out.

A screenshot of a computer screen

Description automatically generated

Regarding edge detection, after trying a number of ways of optimizing the edge detectors, one way which definitely works is the “nothinning” syntax. This improves performance and I know this because I began to see the outline of shapes within the image. I then began experimenting with the threshold values and found that the lower I set the threshold, the brighter the image of segmentation would be. I couldn’t get the Canny operator to work as effectively as the other two.