

Digital Image and Video Processing

Lab 1: Pixel Operations

Student Name:
Hugh Jordan

Student Number:
16321743

Pixel Operations

4. Basic Image Loading and Display

The raw image lenna.256 was loaded using the code provided. This is shown below in Figure 1 and the corresponding histogram is shown in Figure 2. Histograms are a summary of the greyscale distribution of an image.



Figure 1.

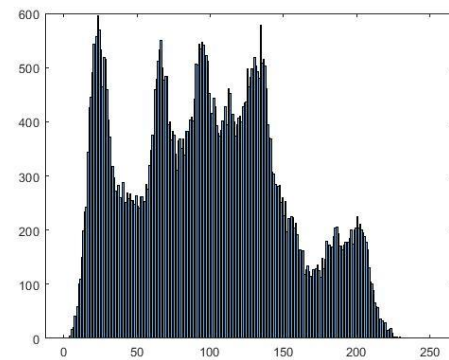


Figure 2.

The line `newpic = pic + 128` was added and the new image was displayed in Figure 3. The new image appears much brighter compared to the original image. This is because the colour values have been offset positively by 128 which has the effect of increasing the brightness and contrast of the picture. This is verified by the corresponding histogram in Figure 4.

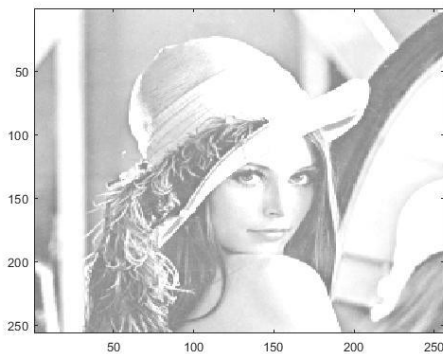


Figure 3.

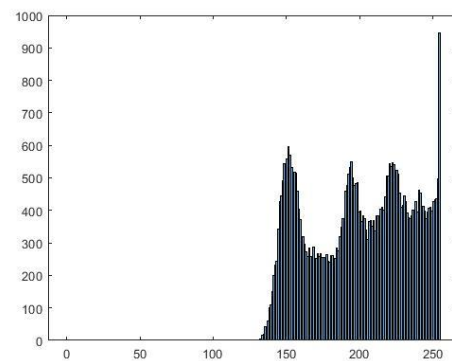


Figure 4.

The line `newpic = pic - 128` was then added and the new image was displayed in Figure 5. Conversely to the previous photo, the new image appears much darker compared to the original image. This is because the colour values have also been offset negatively by 128 which has the effect of decreasing the brightness and contracts of the picture. This is verified by the corresponding histogram in Figure 6.



Figure 5.

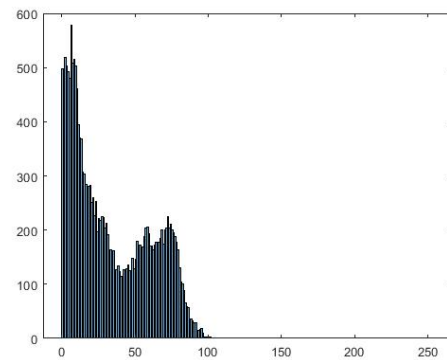


Figure 6.

5 Histograms

The image sigmedia06907.tif was loaded from a user-defined filename and a 2D histogram of the image was generated. The image is shown in Figure 7 and the corresponding histogram is shown in Figure 8.

The line `size(pic)` was then added to the script. The function `d = size(X)` is used to return the size of each of the dimensions of an array `X` in a vector `d`. This means that the line `size(pic)` returns the size of each of the dimensions of the picture.



Figure 7.

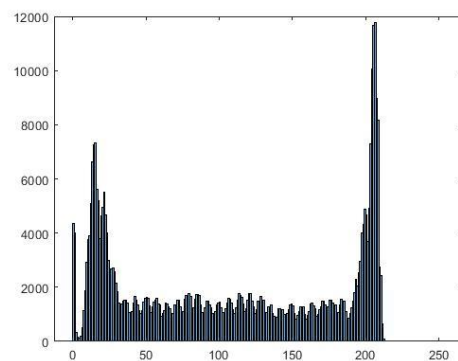


Figure 8.

5.1 Histograms of colour planes

The script above was changed to load the `pool.01.bmp` image instead. The line `[vres,hres]=size(picture)` was added to work out the size of the image. Following this, the three colour planes, red, green and blue were extracted and displayed in Figures 9, 11 and 13 respectively. The corresponding histograms of each of these colour planes were also displayed in Figures 10, 12 and 14 respectively.

The table region of the image is represented by the largest peak of values in each of the three histograms. This means that the table can be isolated by extracting this peak from the red, green and blue channels and concatenating them together afterward. This will be explored in more detail in section 6.

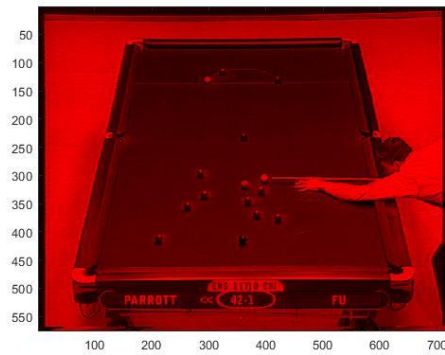


Figure 9.

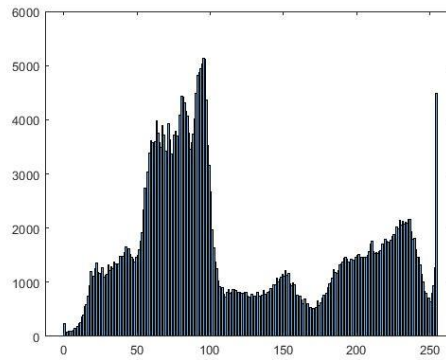


Figure 10

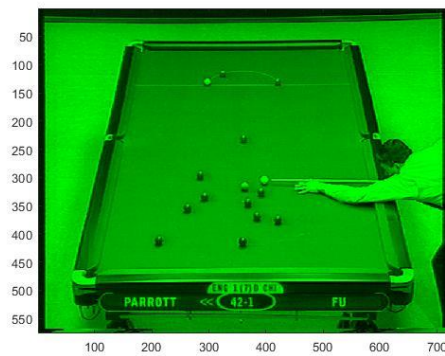


Figure 11.

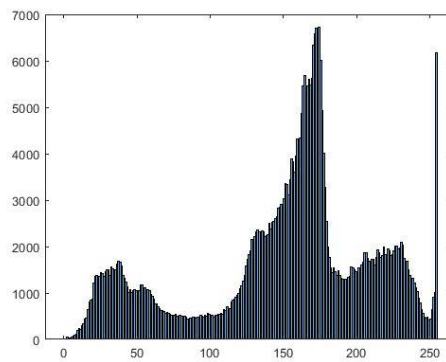


Figure 12.

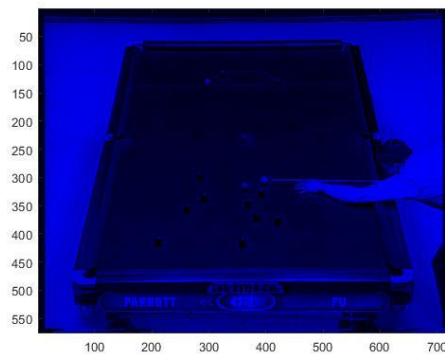


Figure 13.

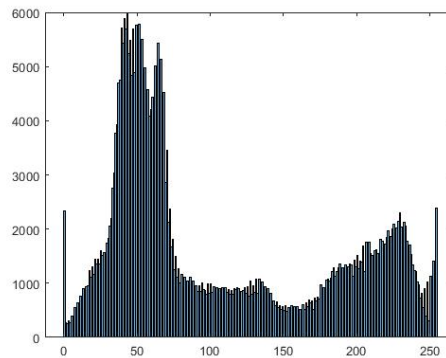


Figure 14.

6. Thresholding

The image was loaded from the file nuclei.02.png and then displayed in figure 15 alongside the image's corresponding histogram in figure 16. The values of t_1 and t_2 were then chosen by examining the histogram and selecting values that encompassed the peak of white greyscale values at the end of the histogram. This meant that the mask would return 1 where the nuclei exist and 0 where they did not. This mask was verified by displaying it in figure 17 and comparing it to the original image. It is evident from figure 15 that this mask correctly identified each of the nuclei.

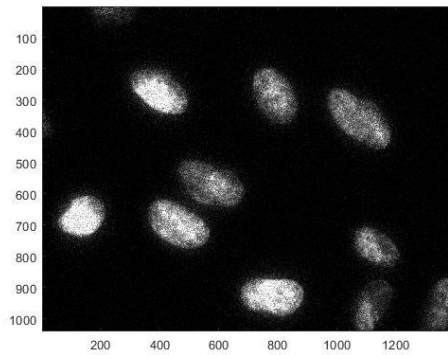


Figure 15.

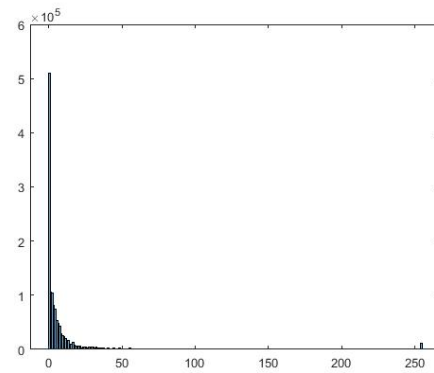


Figure 16.

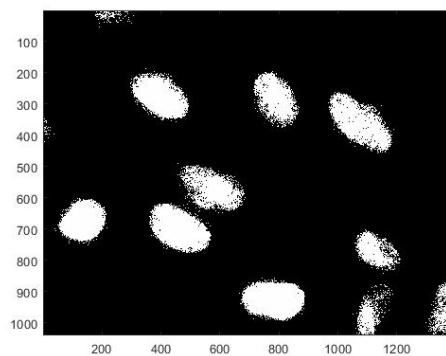


Figure 17.

The histograms of the colour planes from Section 5.1 were then examined and the table was segmented by extracting the largest peak of values from each of the three colour channels and concatenating them all together. The largest peak of values was extracted from each of these colour channels because the table takes up most of the image which means that this peak represents the colour of the table. The segmented table is shown in Figure 17. The algorithm that segments the table successfully isolates the table, but it fails to isolate the table alone. It is evident from Figure 17 that there are segments of green above and below the table which the algorithm failed to ignore. The values of t_1 and t_2 that this algorithm uses were chosen by examining each of the histograms and choosing values that would only encompass the largest peak of values in each of the colour channels. These values are shown in the table below:

	T1	T2
Red	50	100
Green	110	185
Blue	33	72

To demonstrate why applying thresholds on all three colour channels improves segmentation compared to performing the segmentation on the green channel alone, the red, green and blue channels were all segmented alone and displayed in Figure 19 – 21 respectively. It is evident from these figures that despite the table being primarily green, there are elements of the red and blue channels that are present. This creates a significant difference in colour between the segmentation involving all 3 colour channels, as shown in Figure 18, compared to segmentation involving the green

channel alone, as shown in Figure 20. It is clear from these figures that the segmentation involving all 3 colour channels is more colour accurate. The same logic can be applied to any other photo.

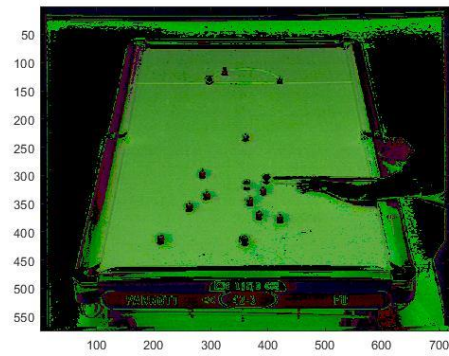


Figure 18.

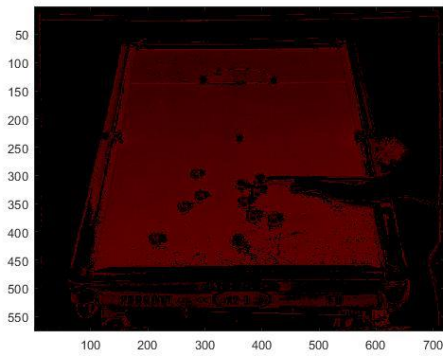


Figure 19.

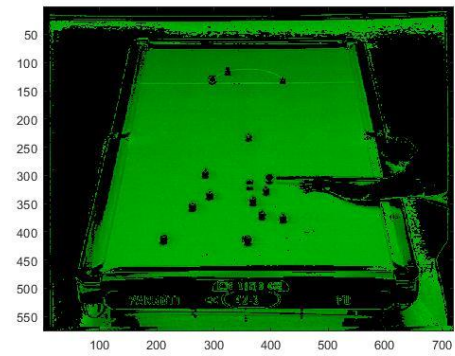


Figure 20.

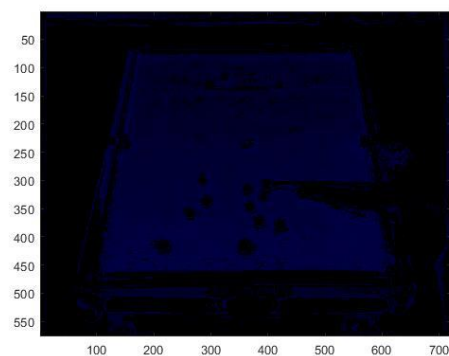


Figure 21.