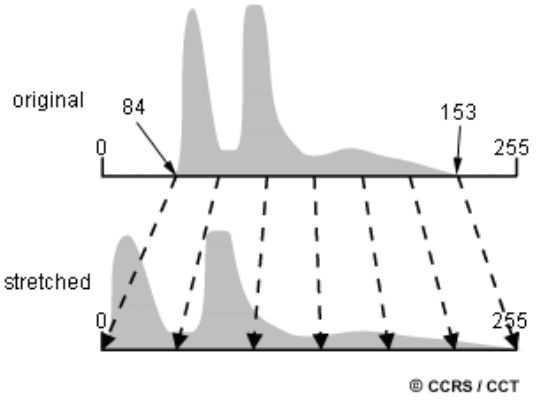
Lab Report 3

Aim

Performing histogram Equalization on images by developing algorithm for it.

Theory

Histogram Equalization



- Histogram Equalization is a computer image processing technique used to improve contrast in images. It accomplishes this by effectively spreading out the most frequent intensity values, i.e. stretching out the intensity range of the image. This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of lower local contrast to gain a higher contrast.

Figure 1: graphical depiction of histogram equalization.

- Histogram equalization is a technique used in image processing to enhance the contrast of an image. It works by redistributing the pixel intensities in an image so that they are more evenly distributed across the full range of intensity values.
- The basic idea behind histogram equalization is to compute a histogram of the pixel intensities in the image, and then to use that histogram to create a mapping function that will transform the original image so that it has a more uniform distribution of pixel intensities.
- The mapping function used in histogram equalization is typically a cumulative distribution function (CDF), which represents the cumulative frequency of each intensity value in the histogram. The CDF is then normalized so that its values range from 0 to 1, and this normalized CDF is used to map the pixel intensities in the original image to new intensity values that are more evenly distributed.
- The result of histogram equalization is an image with improved contrast, where the darker and lighter areas are more pronounced, making it easier to distinguish details in the image. However, it is important to note that histogram equalization may also introduce artifacts or noise in the image if not applied properly, and it may not be appropriate for all types of images.

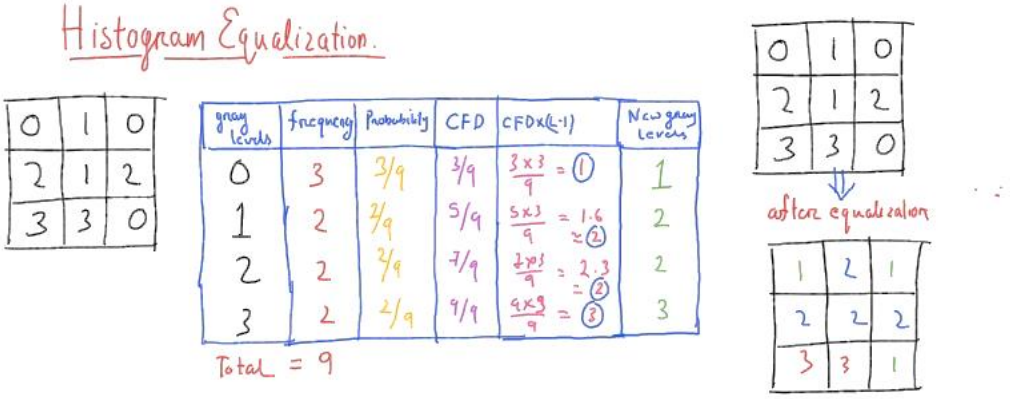


Figure 2: process of finding histogram equalization.

CODE

```
y=uigetfile('*.*');
j=imread(y);
i=rgb2gray(j);
rows=height(i);
column=width(i);
histvalue=zeros(1,256);
for Rows =1:rows
   for Columns=1:column
       x=i(Rows,Columns);
       histvalue(1,x+1)=histvalue(1,x+1)+1;
   end
end
%histogram ends here-----------------------------------
%probability ------------------------------------------
px=zeros(1,256);
for columns=1:256
    px(1,columns)=histvalue(1,columns)/(rows*column);
end
%cfd finding ------------------------------------------
cdf=zeros(1,256);
cumulative=0;
for columns=1:256
    cdf(1,columns)=px(1,columns)+cumulative;
    cumulative=cumulative+px(1,columns);
end
%cfd normalising --------------------------------------
CDF=255*cdf;
newhist=round(CDF);
NEWHIST=zeros(1,256);
for elements=1:256
    newgraylevel=newhist(1,elements)+1;

NEWHIST(1,newgraylevel)=NEWHIST(1,newgraylevel)+histvalue(1,elements);
end


new=histeq(i);
```

```
histn=imhist(new);


figure();
k=0:1:255;
subplot(2,2,1);bar(k,imhist(i));title('Histogram using imhist function')
subplot(2,2,2);bar(k,histvalue);title('Histogram using custom code')
subplot(2,2,3);bar(histn);title('Histogram eualization using histeq function')
subplot(2,2,4);bar(k,NEWHIST);title('Histogram eualization using custom code')
```

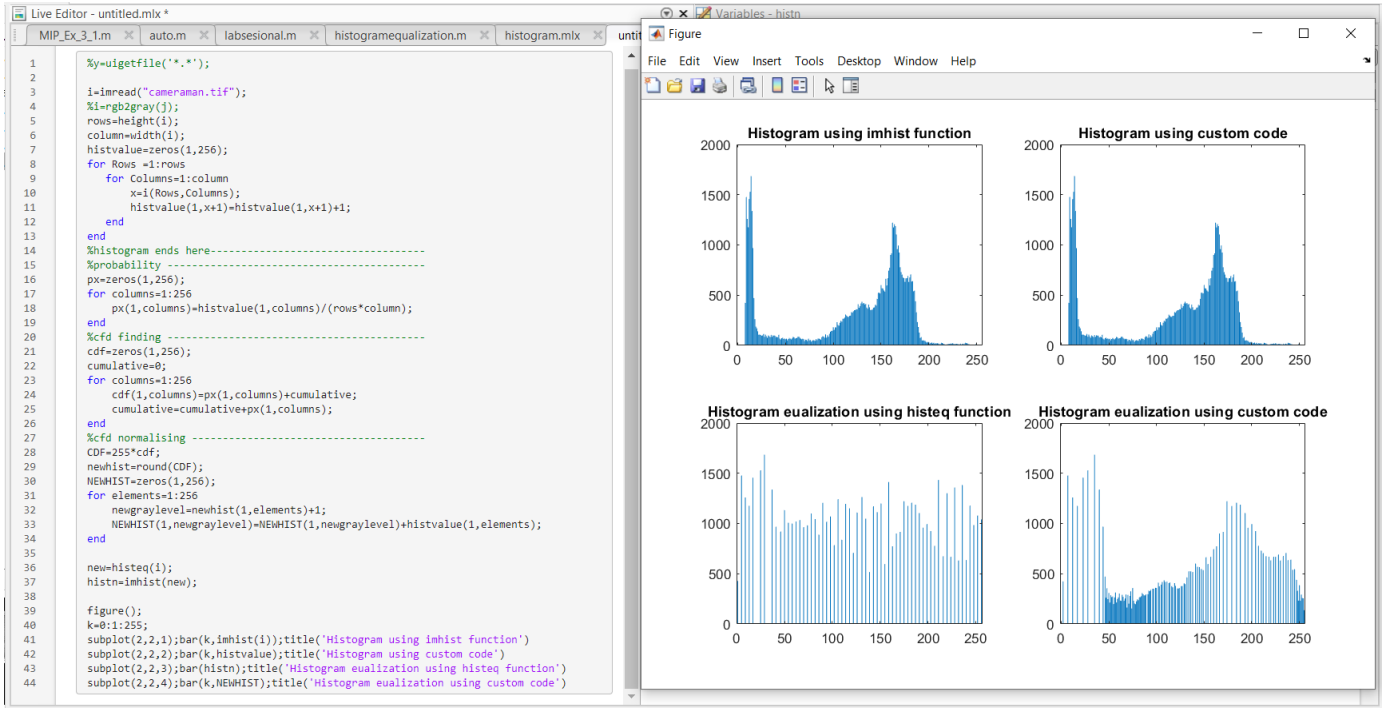


Figure 3: plots showing histograms before and after equalization.

Shreenandan Sahu |120BM0806