CS-663 Assignment 1 Q3

Soham Naha (193079003)
Akshay Bajpai (193079002)
Mohit Agarwala (19307R004)

September 1, 2020

3 Problem Statement

Consider a (non-discrete) image I(x) with a continuous domain and real- valued intensities within [0, 1]. Let the image histogram be h(I), with mass 1. Consider the histogram h(I) is split into two histograms (i) $h_1(I)$ over the domain [0, a] and (ii) $h_2(I)$ over the domain (a, 1], for some arbitrary $a \in (0, 1)$. Assume that the histogram mass within [0, a] is $\alpha \in (0, 1)$

3.1 (8 points)

(a) Suppose you perform histogram equalization over the two intensity in- tervals [0, a] and (a, 1] separately, in a way that preserved the masses of the two histograms $h_1(I)$ and $h_2(I)$ after the transformation. Derive the mean intensity for the resulting histogram (or, equivalently, image) and include it in the report.
Given-

$$\int_0^1 h(I)\ dI = 1 \tag{1}$$

$$\int_0^a h_1(I)\ dI = \alpha = \int_0^a h_1'(I)\ dI \tag{2}$$

$$\int_a^1 h_2(I)\ dI = 1 - \alpha = \int_a^1 h_2'(I)\ dI \tag{3}$$

where $\alpha \in (0, 1)$. In (1), (2) and, (3), h(I) represents the histogram of the image I(x). h(I) is split into two histograms at an intensity $I_1 = a$ resulting in two new image histograms $h_1(I)$ and $h_2(I)$. $h_1'(I)$ and h_2' (I) are the histograms after performing histogram equalization.
To find- Mean intensity of the resulting histogram after transformation

$$E'[I] = \int_0^1 Ih'(I)\ dI = \int_0^a Ih_1'(I)\ dI + \int_a^1 Ih_2'(I)\ dI = ? \tag{4}$$

Solution-
Let the histogram values after equalization be h_1' and h_2' respectively. Since after equalization, these values can be assumed to be constant, Using (2) and (3) with (4) we get:
$\alpha = \int_0^a h_1'(I)\ dI = h_1' \int_0^a dI = h_1' * a$
So, $h_1' = \frac{\alpha}{a}$
$1 - \alpha = \int_a^1 h_2'(I)\ dI = h_2' \int_a^1 dI = h_2' * (1 - a)$
So, $h_2' = \frac{1-\alpha}{1-a}$

$$\begin{aligned}
E'[I] &= \int_0^1 Ih'(I)\ dI \\
&= \int_0^a Ih_1'(I)\ dI + \int_a^1 Ih_2'(I)\ dI \\
&= \frac{h_1' * I^2|_0^a}{2} + \frac{h_2' * I^2|_a^1}{2} \\
&= \frac{\frac{\alpha}{a} * (a^2)}{2} + \frac{\frac{1-\alpha}{1-a} * (1 - a^2)}{2} \\
&= \frac{a * \alpha}{2} + \frac{(1 + a) * (1 - \alpha)}{2} \\
&= \frac{a\alpha + 1 + a - \alpha - a\alpha}{2} \\
&= \frac{1 + a - \alpha}{2}
\end{aligned} \tag{5}$$

3.2 (2 points)

(b) Let the chosen intensity a be the median intensity for the original his- togram h(I). Assume that the mean intensity for the original histogram h(I) is also a. Then, what is the mean intensity for the resulting histogram (or, equivalently, image). Show the derivations clearly in the report.

Given-

$$\int_0^a h(I)\ dI = \int_a^1 h(I)\ dI = \frac{1}{2} \rightarrow \alpha = \frac{1}{2} \tag{6}$$

$$E[I] = \int_0^1 Ih(I)\ dI = a \tag{7}$$

To find-

$$E'[I] = \int_0^1 Ih'(I)\ dI = \int_0^a Ih_1'(I)\ dI + \int_a^1 Ih_2'(I)\ dI = ? \tag{8}$$

Solution-

$$E'[I] = \frac{1+a-\alpha}{2}|_{\alpha=1/2} = \frac{0.5+a}{2} \tag{9}$$

3.3 (5 points)

(c) Describe a scenario where the above described histogram-based intensity transform with a $= I_{Median}$ will do a better job in intensity transformation than a simple histogram equalization. Explain the reasons clearly.

Solution- We apply the standard histogram equalization algorithm on the image histogram h(I) of the image I(x) with the following property:

$$\int_0^1 h(I)\ dI = 1 \tag{10}$$

where the initial mean $E[I]$ is also taken to be as a.

Let the value of the transformed histogram $h'(I)$ after the standard equalization be h' which is a constant. Then using conservation of mass,

$$\int_0^1 h(I)\ dI = \int_0^1 h'(I)\ dI = h' * I|_0^1 = 1 \rightarrow h' = 1 \tag{11}$$

Also,

$$E'[I] = \int_0^1 Ih'(I)\ dI = \int_0^1 I * h'\ dI = \frac{I^2|_0^1}{2} = \frac{1}{2} \tag{12}$$

The mean after equalization is always going to be equal to $\frac{1}{2}$ independent of the image.

On the other hand, if we perform histogram equalization using the conditions given in **(b)** the results differ from (12). In this case the mean $E'[I]$ can be written as: $E'[I] = \frac{1+a-\alpha}{2}$

To preserve the brightness of the image, we want to minimize the distance between the initial mean a and final mean $E'[I]$ for which:

$$min \quad d(a, E'[I]) = min \quad |a - E'[I]| = min_{\alpha \in (0,1)} \quad |a - \frac{1+a-\alpha}{2}| \tag{13}$$

which on optimizing, results in $\alpha = \frac{1}{2}$ and thus concluding that a should be the median of the image histogram $h(I)$,

$$\int_0^a h(I)\ dI = \alpha = \frac{1}{2} \tag{14}$$

which takes the mean of the finally transformed image histogram closer to the initial mean a. In this case the mean value $E''[I]$ can be given from eq (9) to be $E''[I] = \frac{0.5+a}{2}$. This might be a good approach if we wish to keep the image's overall brightness intact.

3.4 (10 points)

(d) Do an online search to find an image along the lines of your reasoning. Write a code for this intensity transformation and demonstrate the better performance on the image you obtained. Note that the better performance should be distinctly evident.

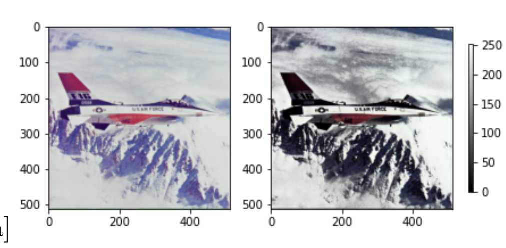
Solution-

```python
import cv2
import numpy as np
import matplotlib.pyplot as plt
from seaborn import distplot

def plot_hist(input_file,input_image,output_image):
    """
    input : input_file_path, input_image, output_image
    output : saves the histograms for both the images for comparison
    dependencies : seaborn, numpy, matplotlib
    """
    name = input_file.split(".")[2]
    plt.figure()
    plt.title("Normalized Histogram Plots for Images")
    ax = distplot(input_image,color='r',label ="Input Histogram",
        hist_kws={"alpha": 0.3, "linewidth": 1.5},bins=256,hist=False)
    ax = distplot(output_image,color="b",label ="BiHistogram Histogram",
        hist_kws={"alpha": 0.3,"linewidth": 1.5},bins=256,hist=False)
    l1 = ax.lines[0]
    x1 = l1.get_xydata()[:,0]
    y1 = l1.get_xydata()[:,1]
    ax.fill_between(x1,y1, color="red", alpha=0.3)
    l2 = ax.lines[1]
    x2 = l2.get_xydata()[:,0]
    y2 = l2.get_xydata()[:,1]
    ax.fill_between(x2,y2, color="blue", alpha=0.3)
    plt.legend()
    plt.savefig(".."+name+"BiHistHistogram.png",bbox_inches="tight",pad=-1)

def mySubHE (input_file):

    name = input_file.split(".")[2]
    image = cv2.imread(input_file)
    image = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)

    hsv_image = cv2.cvtColor(image,cv2.COLOR_RGB2HSV)
    h,s,v = cv2.split(hsv_image)

    rows,columns=image.shape[:2]

    output_image = v.copy();
    intensity_freq = np.zeros((256,1))
    intensity_prob = np.zeros ((256,1))
    intensity_cdf  = np.zeros((256,1))

    for i in range(rows):
        for j in range(columns):
            x=int(v[i,j])
            if x>255:
                x=255
            intensity_freq[x]=intensity_freq[x]+1


    mean_val=np.mean(v)
    median_val=np.median(v)

    lastval=int(median_val)
    firstval=lastval+1

    subh1= intensity_freq[0:int(lastval)+1]
    subh2 = intensity_freq[int(firstval) : 256 ]

    cdf1 = equalize(v,output_image,subh1,0,lastval)
    cdf2 = equalize(v,output_image,subh2,firstval,255)


    for i in range(rows):
        for j in range(columns):
            x=int(v[i,j])
            if x>255:
                x=255

            if x<=lastval:
                output_image[i,j]= 1+ (lastval-1)*cdf1[x]
            elif x>lastval:
```

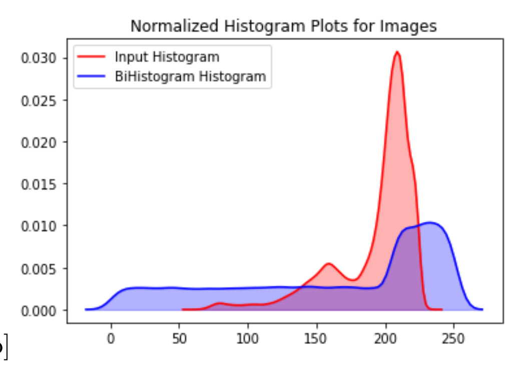
```
                output_image[i,j]= firstval+(255-firstval)*cdf2[x-lastval]

    plot_hist(input_file,v,output_image)
    hsv_image[:,:,2] = output_image
    output_image = cv2.cvtColor(hsv_image,cv2.COLOR_HSV2RGB)

    fig,axes = plt.subplots(1,2, constrained_layout=True, gridspec_kw={'width_ratios':[1,1]})
    axes[0].imshow(image,cmap="gray")
    axes[0].axis("on")
    im = axes[1].imshow(output_image,cmap="gray")
    axes[1].axis("on")
    plt.imshow(output_image,cmap="gray")
    cbar = fig.colorbar(im,ax=axes.ravel().tolist(),shrink=0.45)

    plt.savefig(".."+name+"SUBHEcombined.png",bbox_inches="tight",pad=-1)
    plt.imsave(".." +name +"SUBHE.png",output_image,cmap="gray")


def equalize(image,output_image,hist_,val1,val2):
    rows,columns=image.shape

    num_samples = np.sum(hist_)
    intensity_prob = np.zeros((val2-val1+2,1));
    intensity_cdf = np.zeros((val2-val1+2,1));

    for i in range(rows):
        for j in range(columns):
            x=int(image[i,j]);
            if x > val1 and x <=val2:
                intensity_prob[x-val1]= hist_[x-val1]/num_samples;

    a,b=intensity_cdf.shape;
    for j in range(len(intensity_cdf)):
        intensity_cdf[j] = np.sum(intensity_prob[1:j]);
    return intensity_cdf
```

myMainScript.py

```
from myHE import myHE
from mySubHE import mySubHE

file_path = '../data/f16.tiff'
mySubHE(file_path)
myHE(file_path)
```

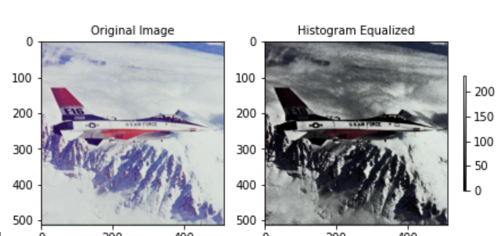


[a]

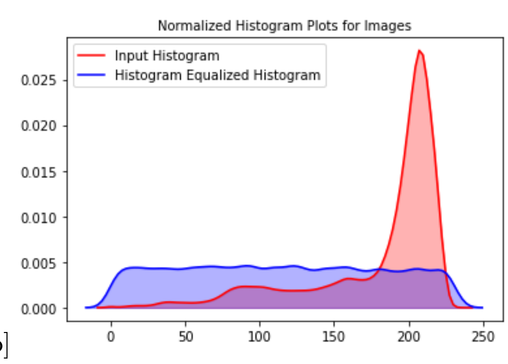


[b]

Figure 1: (a) BiHistogram Equalization of F16.tiff (b) Histogram comparison for (a)

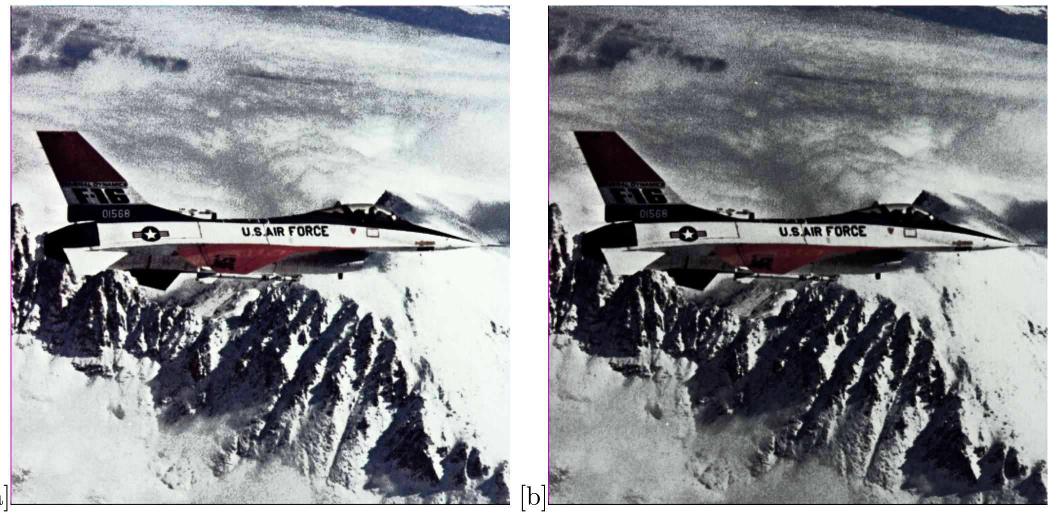


[a]



[b]

Figure 2: (a) Histogram Equalization of F16.tiff (b) Histogram comparison for (a)



[a] [b]

Figure 3: (a) Bi-Histogram Equalization of F16.tiff (b) Histogram Equalization of F16.tiff

Observation:
We see that both Histogram Equalization(HE) and Bi-Histogram equalization (Bi-HE) increases the contrast of the original image. But the performance of Bi-HE is better as it preserves the brightness component of the original image which is absent in HE, also visible from the Histogram plots of figure 1(b) and 2(b).