image_informatics_lab_4

November 28, 2022

1 Image Informatics - Lab 4

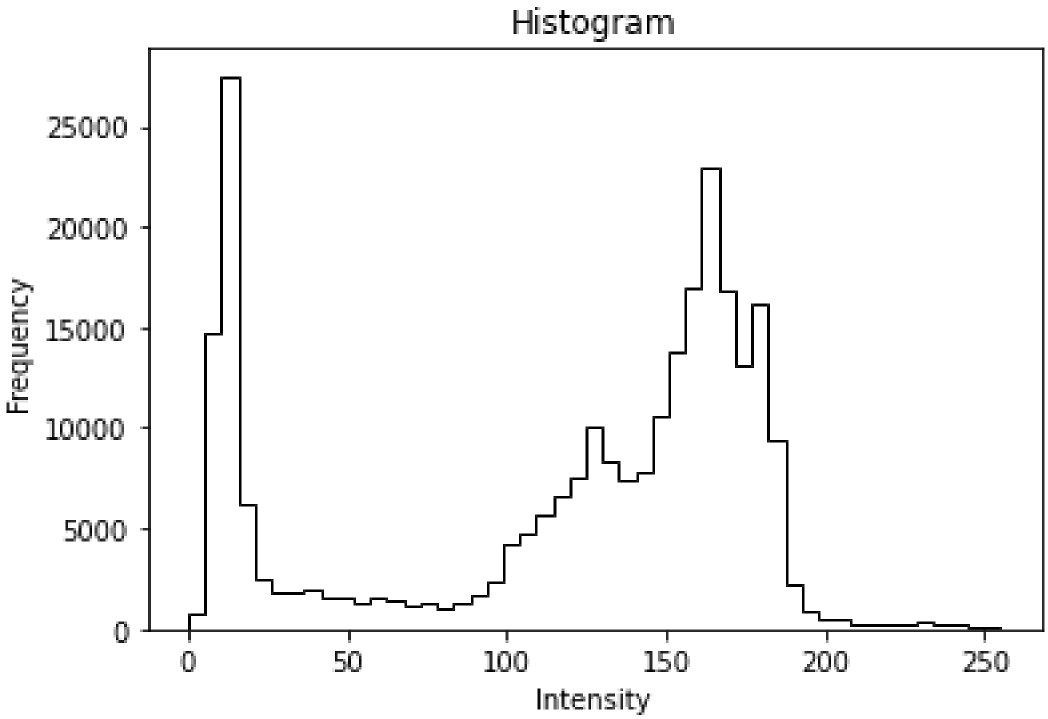
1.1 Histograms & Histogram Processing

Tasks:

1. Plot the histograms of each of the R, G and B colour channels from an image. Plot each in it corresponding colour and overlay them as a single image
2. Perform histogram equalization on each of the R, G and B colour channels from an image.

[6]:

```
import numpy as np
from skimage import data
import matplotlib.pyplot as plt

im = data.camera()
v = np.concatenate(im)
bins = np.linspace(0, 255, 50)
p = plt.hist(v, bins, color='k', histtype='step')
t = plt.title("Histogram")
t = plt.xlabel('Intensity')
t = plt.ylabel('Frequency')
```



1.2 Colour Image Processing

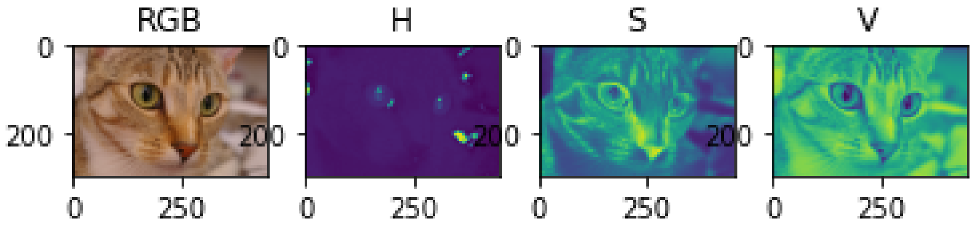
Tasks:

1. Separate out objects in the scene of a particular Hue value. Add these back as a colour overlay onto the original image in grayscale as per the example of the rose (Hue = Red).
2. Perform Histogram Equalization on the V channel of an HSV version of a RGB colour image before converting back to RGB for display. Compare this to just performing Histogram Equalization on each channel of the RGB version of he image. Display both version in different windows. The difference in result is also apparent for all contrast enhancement and manipulation approaches.

[3]:

```
from skimage import data
from skimage.color import rgb2hsv
import matplotlib.pyplot as plt

im = data.chelsea()
im_hsv = rgb2hsv(im)
im_hsv_h = im_hsv[:,:,0]
im_hsv_s = im_hsv[:,:,1]
im_hsv_v = im_hsv[:,:,2]

fig = plt.figure()
ax = fig.add_subplot(1, 4, 1)
```

```
p = plt.imshow(im)
t = ax.set_title('RGB')

ax = fig.add_subplot(1, 4, 2)
p = plt.imshow(im_hsv_h)
t = ax.set_title('H')

ax = fig.add_subplot(1, 4, 3)
p = plt.imshow(im_hsv_s)
t = ax.set_title('S')

ax = fig.add_subplot(1, 4, 4)
p = plt.imshow(im_hsv_v)
t = ax.set_title('V')
```


RGB
H
S
V