Technische Universität München
Photogrammetrie und Fernerkundung
Univ.-Prof. Dr.-Ing. U. Stilla

Exercises in Photogrammetry, Remote Sensing, and Image Processing

Name, Given name:

Ho, Hsin-Feng 03770686

Exercise number.: 1

Topic: Image Characteristics

Study Program: ESPACE

(filled in by supervisor)

Date: ______________

Points: ______________

Supervisor: ______________

2023

Hsin-Feng Ho
03770686

1 Image Characteristics

1.1 Image Histogram

The histogram of an image can be computed by counting the number of pixels with each intensity value. The histogram of an image can be used to get an idea of the contrast of the image, the dynamic range of the intensity values, and the brightness of the image.

```
hist = np.zeros(256)
for i in range(img.shape[0]):
  for j in range(img.shape[1]):
    hist[img[i,j]] += 1
```

The code iterates through the image and counts the number of pixels with each intensity value. The result is shown in Figure 1.

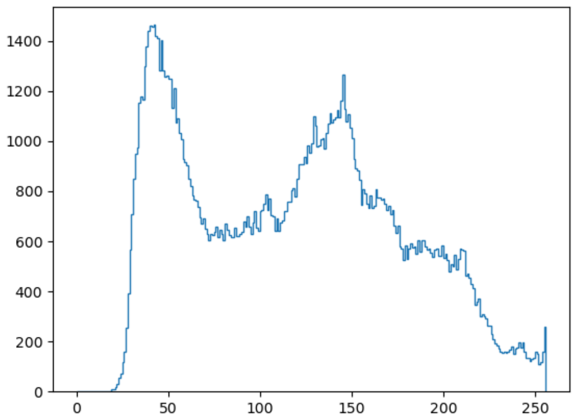


Figure 1: Histogram of the image

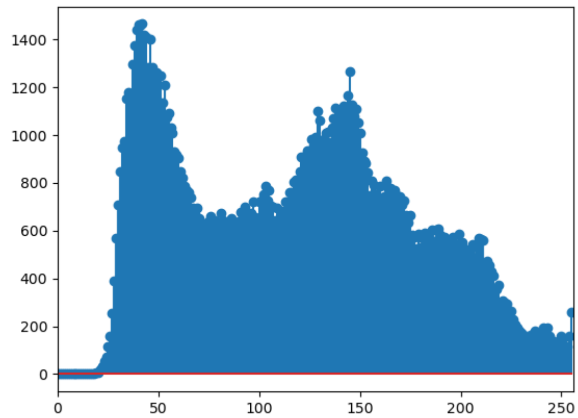


Figure 2: Check the histogram using plt.stem()

1.2 mean, variance, and standard deviation

The mean, variance, and standard deviation of an image can be computed by the following equations:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^2$$

$$\sigma = \sqrt{\sigma^2}$$

Implementing the equations in Python:

```
mean = np.sum(img)/np.size(img)
var = np.sum((img-mean)**2)/np.size(img)
std = np.sqrt(var)
```

The results are shown below and checked with numpy functions.

```
mean =  118.41953125
np.mean =  118.41953125
var =  3413.9199622802735
np.var =  3413.9199622802735
std =  58.42875971882574
np.std =  58.42875971882574
```

2 Correlation coefficient

The correlation coefficient and covariance of two images can be computed by the following equation:

$$\sigma_{xy} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)$$

$$\begin{aligned}\rho &= \frac{\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{N}(x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{N}(y_i - \mu_y)^2}} \\ &= \frac{\sum_{i=1}^{N} x_i y_i - N\mu_x\mu_y}{\sqrt{\sum_{i=1}^{N} x_i^2 - N\mu_x^2}\sqrt{\sum_{i=1}^{N} y_i^2 - N\mu_y^2}}\end{aligned}$$

Implementing the equations in Python:

```
def cov(img1, img2):
    cov = np.zeros((img1.shape[0], img1.shape[1]))
    cov=
       np.sum((img1-np.mean(img1))*(img2-np.mean(img2)))/np.size(img1)
    return cov

corr = cov(img1,img2)/(np.std(img1)*np.std(img2))
```

The results are shown below and checked with numpy functions.

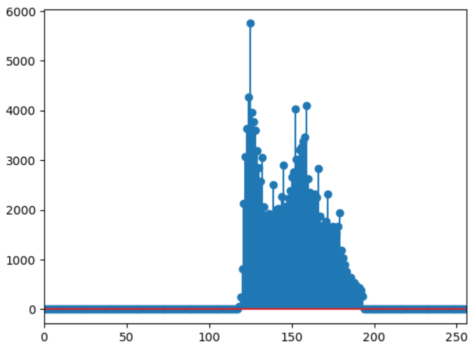
```
cov =  -1084.5263637597654
np.cov =  -1084.5331420919058
corr =  -0.9998783738444597
np.corrcoef =  -0.9998783738444569
```

We can see there is a very slight numerical difference in covariances.

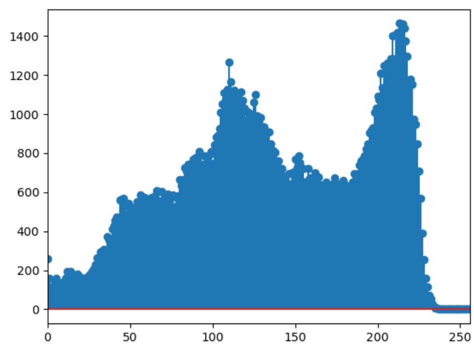
3 Test images

3.1 Histogram, mean, variance, and standard deviation

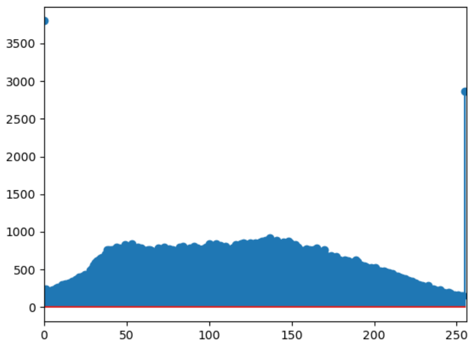
In this task we are going to compute the histograms, mean, variance, and standard deviation of the test images. The results are shown below.



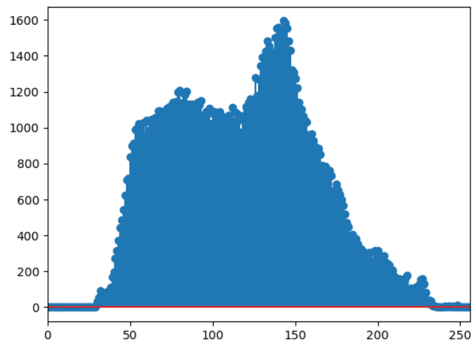
a) Image a



b) Image b



c) Image c



d) Image d

```
Image  images/image_a.bmp :
mean =  149.1242875
var =  344.61370261734373
std =  18.56377393251016
Image  images/image_b.bmp :
mean =  136.58046875
var =  3413.9199622802735
std =  58.42875971882574
Image  images/image_c.bmp :
mean =  118.48776875
var =  4196.281325396523
std =  64.7787104332629
Image  images/image_d.bmp :
mean =  117.9137875
var =  1840.1958299048438
```

```
std =  42.89750377242065
```

3.2 Corvariance and correlation coefficient

In this task we are going to compute the covariance and correlation coefficient of the test images. The results are shown below.

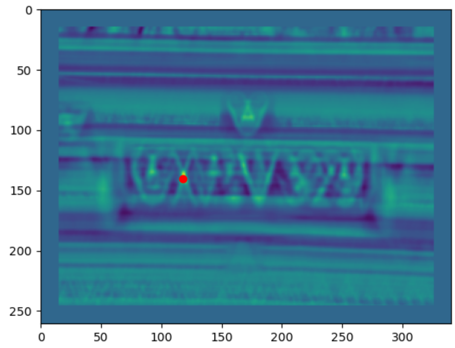
```
Image  images/image_a.bmp :
cov =  1084.5263637597654
corr =  0.9998783738444597
Image  images/image_b.bmp :
cov =  -3413.9199622802735
corr =  -1.0000000000000002
Image  images/image_c.bmp :
cov =  3343.296459516602
corr =  0.8833156452953814
Image  images/image_d.bmp :
cov =  2145.6104938378908
corr =  0.8560363290240045
```

The correlation coefficient demenstrates how similar the two compared images and the range of the correlation coefficient lies between 1 to -1. The 1 stands for the two images are identical and the -1 stands for the two images are completely different. The covariance is the measure of how much two random variables change together. The positive covariance means the two variables are positively related and the negative covariance means the two variables are negatively related.

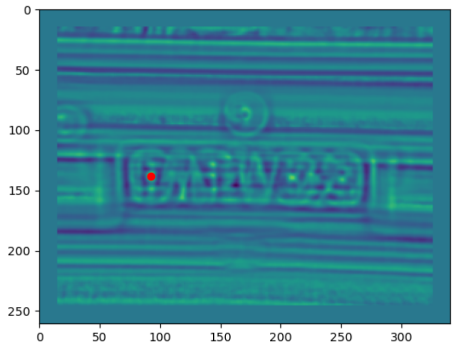
So we can summerize the results that the image a, c and d are very similar to the original image and the image b is the inverted image.

4 Template search

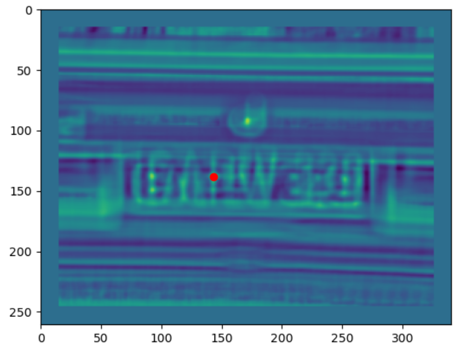
In this task we are going to find the template in the image using the correlation coefficient. The results are shown below.



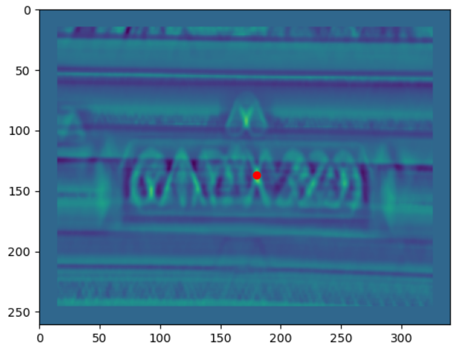
a) Search on A



b) Search on G



c) Search on P



d) Search on V

The principle of a template search is to find the most similar part of the image to the template. The correlation coefficient is used to measure the similarity between the template and the image.