Aim: Apply GLCM To Determine The Texture Of A Given Image‘

IPPR Lab 10

Name : Arya Shah

Roll No. E071

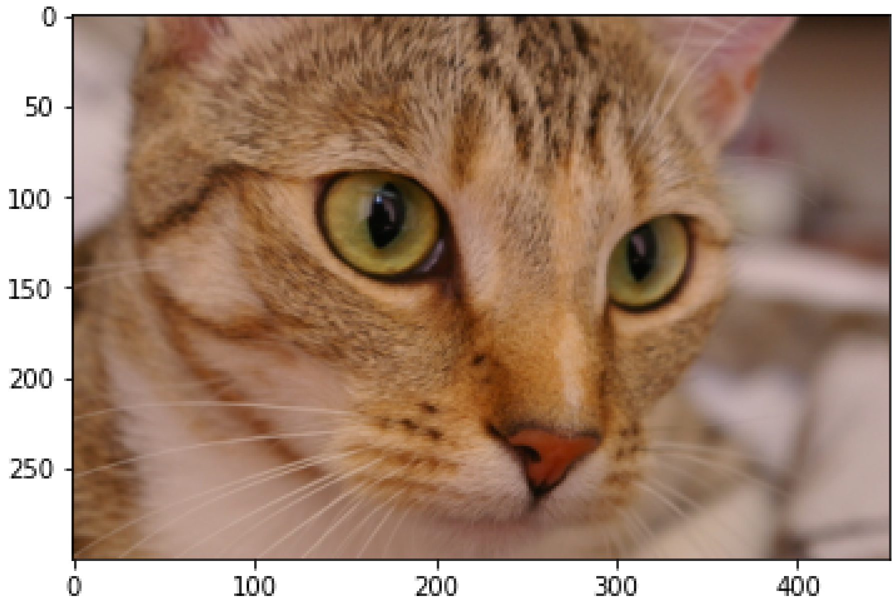
Class : BTech CSBS

```
# Importing Libraries
from skimage import io
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
from skimage.color import rgb2gray, rgba2rgb
import cv2
from skimage import data
from skimage.feature import graycomatrix, graycoprops
```

```
image_c = data.cat()
```

```
plt.imshow(image_c)
```

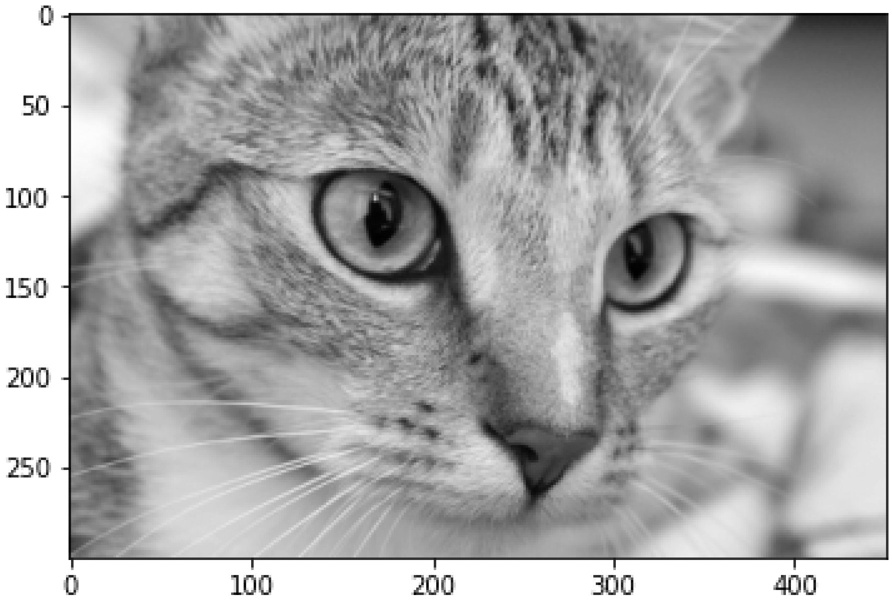
```
<matplotlib.image.AxesImage at 0x264f560a208>
```



```
image = rgb2gray(image_c)
```

```
plt.imshow(image, cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x264f56d9ac8>
```



```
image=image*255
```

```
image.shape
```

```
(300, 451)
```

```
image[0,0]
```

```
123.73389999999999
```

```
image = image.astype('uint8')
```

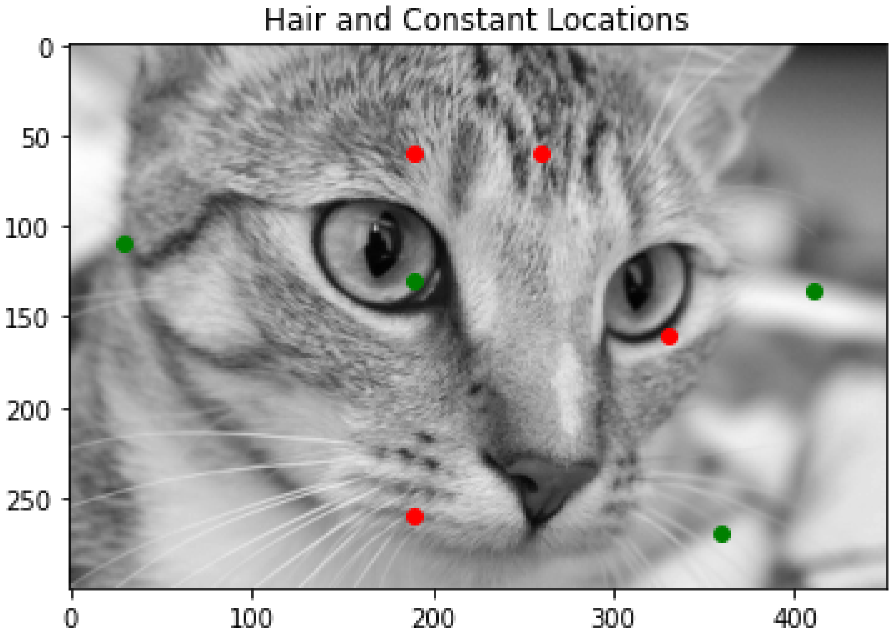
```
image[0,0]
```

```
123
```

```
sz = 20
# Whisker locations
hair_locations = [(50,250),(50,180),(250,180),(150,320)]
constant_locations = [(100,20),(260,350),(120,180),(125,400)]
```

```
plt.imshow(image, cmap = 'gray')
plt.title('Hair and Constant Locations')
for c,r in hair_locations:
    plt.plot(r+(sz/2),c+(sz/2),'ro')
```

```
for c,r in constant_locations:
    plt.plot(r+(sz/2),c+(sz/2),'go')
```



```
hair_image = []
for i in hair_locations:
    hair_image.append(image[i[0]:i[0]+sz,i[1]:i[1]+sz])
```

```
len(hair_image)
```

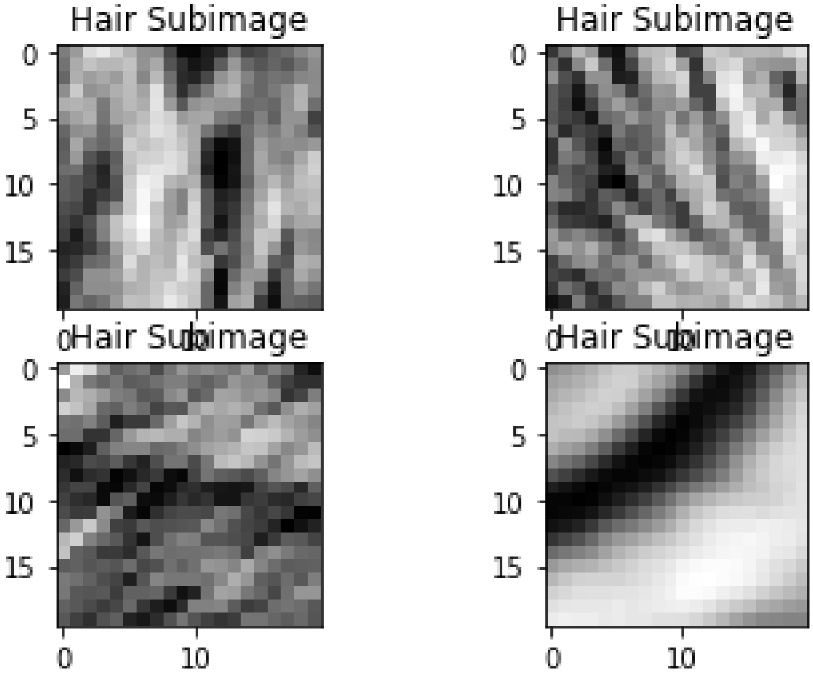
```
4
```

```
constant_image = []
for i in constant_locations:
    constant_image.append(image[i[0]:i[0]+sz,i[1]:i[1]+sz])
```

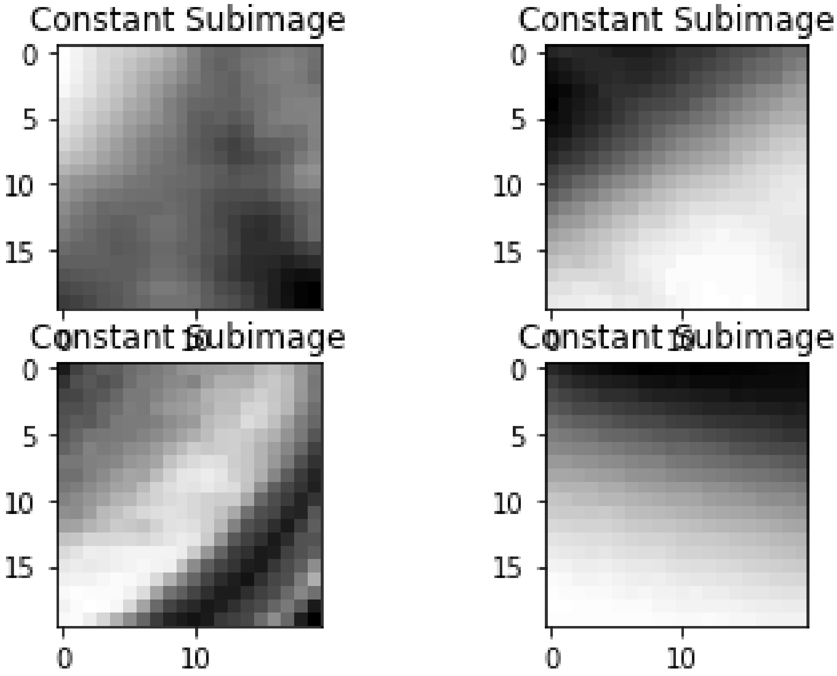
```
len(constant_image)
```

```
4
```

```
for i, subimage in enumerate(hair_image):
    plt.subplot(2,2,i+1)
    plt.imshow(subimage, cmap='gray')
    plt.title('Hair Subimage')
```



```
for i, subimage in enumerate(constant_image):
    plt.subplot(2,2,i+1)
    plt.imshow(subimage, cmap='gray')
    plt.title('Constant Subimage')
```



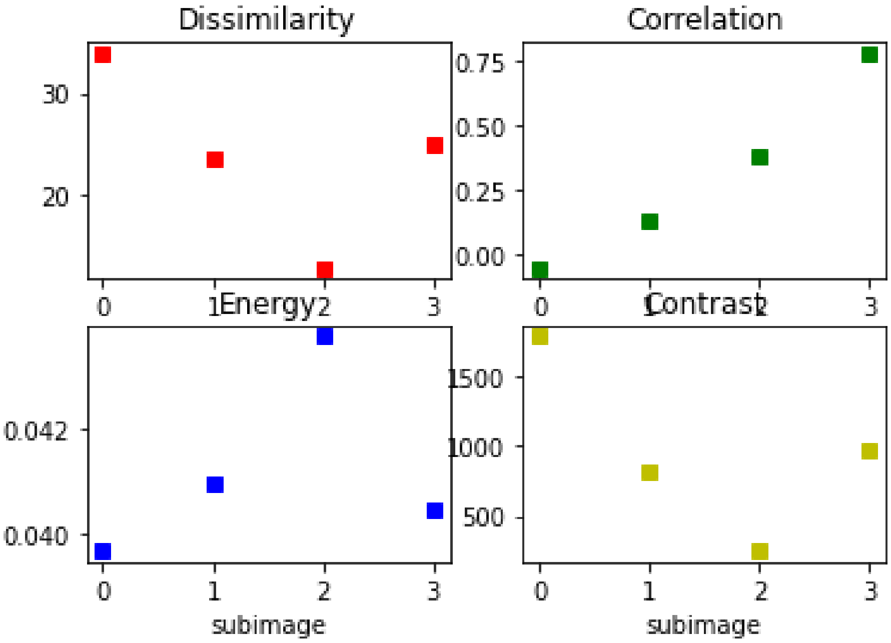
```
diss = []
corr = []
ener = []
cont = []

for i in hair_image:
    glcm = graycomatrix(i, distances = [3], angles = [0],
                        levels = 256, symmetric = True, normed = True)
    diss.append(graycoprops(glcm, 'dissimilarity')[0,0])
    corr.append(graycoprops(glcm, 'correlation')[0,0])
    ener.append(graycoprops(glcm, 'energy')[0,0])
    cont.append(graycoprops(glcm, 'contrast')[0,0])
```

```
len(diss)
```

```
4
```

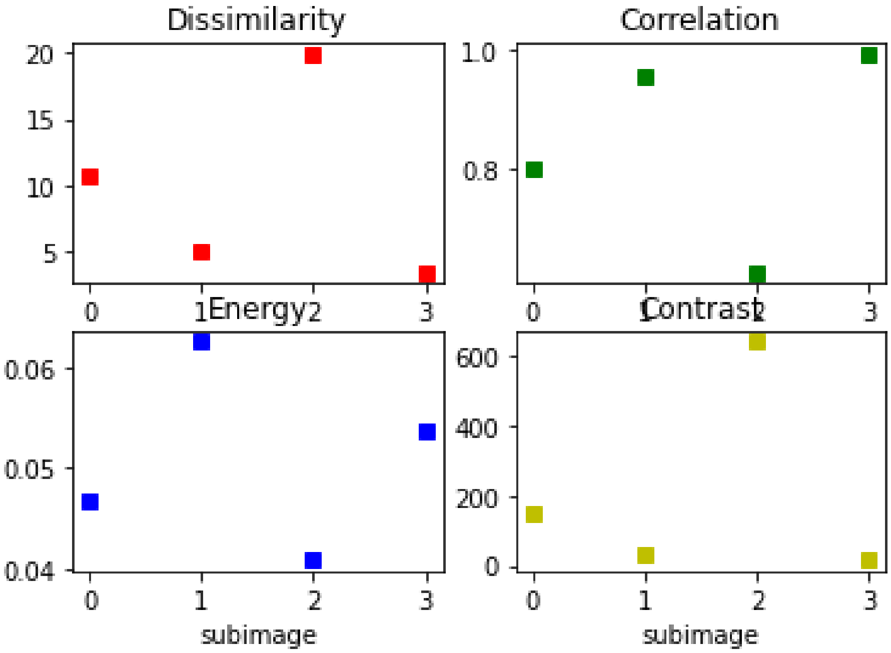
```
diss1 = []
corr1 = []
ener1 = []
cont1 = []
for i in constant_image:
    glcm = graycomatrix(i, distances = [3], angles = [0],
                       levels = 256, symmetric = True, normed = True)
    diss1.append(graycoprops(glcm, 'dissimilarity')[0,0])
    corr1.append(graycoprops(glcm, 'correlation')[0,0])
    ener1.append(graycoprops(glcm, 'energy')[0,0])
    cont1.append(graycoprops(glcm, 'contrast')[0,0])

plt.subplot(2,2,1)
plt.plot(diss,'rs')
plt.title('Dissimilarity')
plt.xlabel('subimage')

plt.subplot(2,2,2)
plt.plot(corr,'gs')
plt.title('Correlation')
plt.xlabel('subimage')

plt.subplot(2,2,3)
plt.plot(ener,'bs')
plt.title('Energy')
plt.xlabel('subimage')

plt.subplot(2,2,4)
plt.plot(cont,'ys')
plt.title('Contrast')
plt.xlabel('subimage')
```

```
Text(0.5, 0, 'subimage')
```



```
# For Constant Image
plt.subplot(2,2,1)
plt.plot(diss1,'rs')
plt.title('Dissimilarity')
plt.xlabel('subimage')

plt.subplot(2,2,2)
plt.plot(corr1,'gs')
plt.title('Correlation')
plt.xlabel('subimage')

plt.subplot(2,2,3)
plt.plot(ener1,'bs')
plt.title('Energy')
plt.xlabel('subimage')

plt.subplot(2,2,4)
plt.plot(cont1,'ys')
plt.title('Contrast')
plt.xlabel('subimage')
```

```
Text(0.5, 0, 'subimage')
```



Conclusion

- A higher dissimilarity value indicates that neighboring pixels are dissimilar in their gray-level values while A high correlation value indicates that neighboring pixels have similar gray-level values.
- A high energy value indicates that gray-level values are evenly distributed in the image while higher contrast value indicates that there is a greater difference in gray-level values between neighboring pixels.
- dissimilarity and contrast are inversely related, meaning that when one increases, the other decreases.
- Similarly, correlation and energy are directly related, meaning that when one increases, the other increases as well.

Aim: Apply GLCM To Determine The Texture Of A Given Image‘

IPPR Lab 10

Name : Arya Shah

Roll No. E071

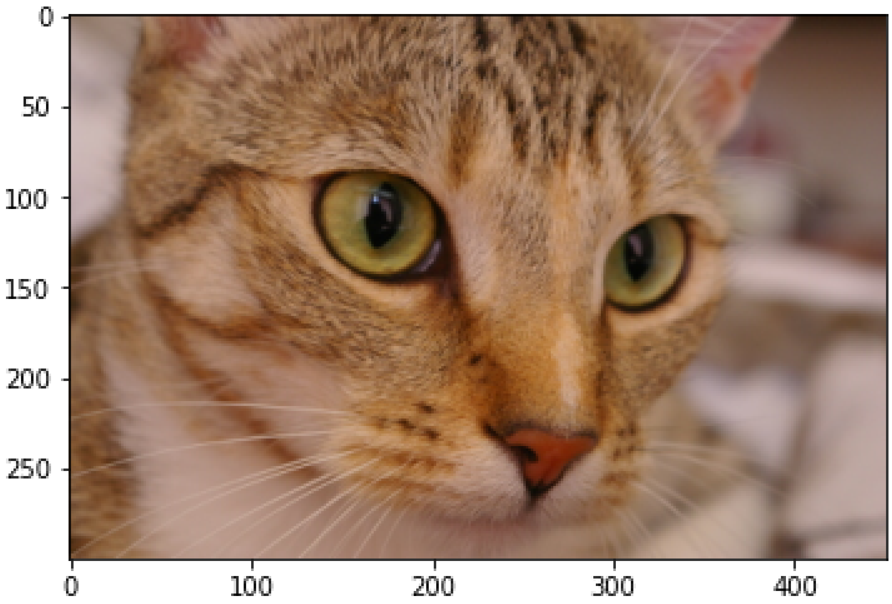
Class : BTech CSBS

```
# Importing Libraries
from skimage import io
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
from skimage.color import rgb2gray, rgba2rgb
import cv2
from skimage import data
from skimage.feature import graycomatrix, graycoprops
```

```
image_c = data.cat()
```

```
plt.imshow(image_c)
```

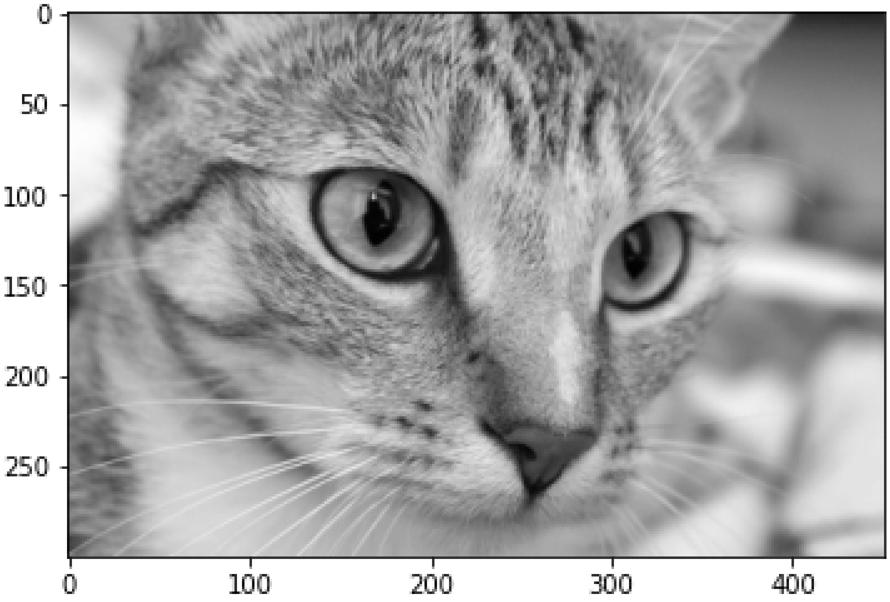
```
<matplotlib.image.AxesImage at 0x20dfc7ba508>
```



```
image = rgb2gray(image_c)
```

```
plt.imshow(image, cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x20dfc88f388>
```



```
image=image*255
```

```
image.shape
```

```
(300, 451)
```

```
image[0,0]
```

```
123.73389999999999
```

```
image = image.astype('uint8')
```

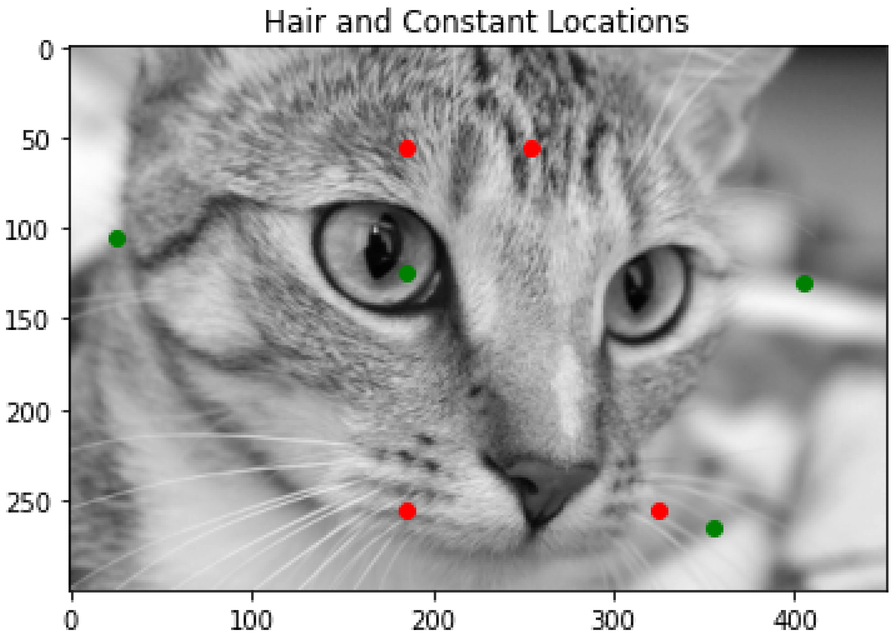
```
image[0,0]
```

```
123
```

```
sz = 10
# Whisker locations
hair_locations = [(50,250),(50,180),(250,180),(250,320)]
constant_locations = [(100,20),(260,350),(120,180),(125,400)]
```

```
plt.imshow(image, cmap = 'gray')
plt.title('Hair and Constant Locations')
for c,r in hair_locations:
    plt.plot(r+(sz/2),c+(sz/2),'ro')
```

```
for c,r in constant_locations:
    plt.plot(r+(sz/2),c+(sz/2),'go')
```



```
hair_image = []
for i in hair_locations:
    hair_image.append(image[i[0]:i[0]+sz,i[1]:i[1]+sz])
```

```
len(hair_image)
```

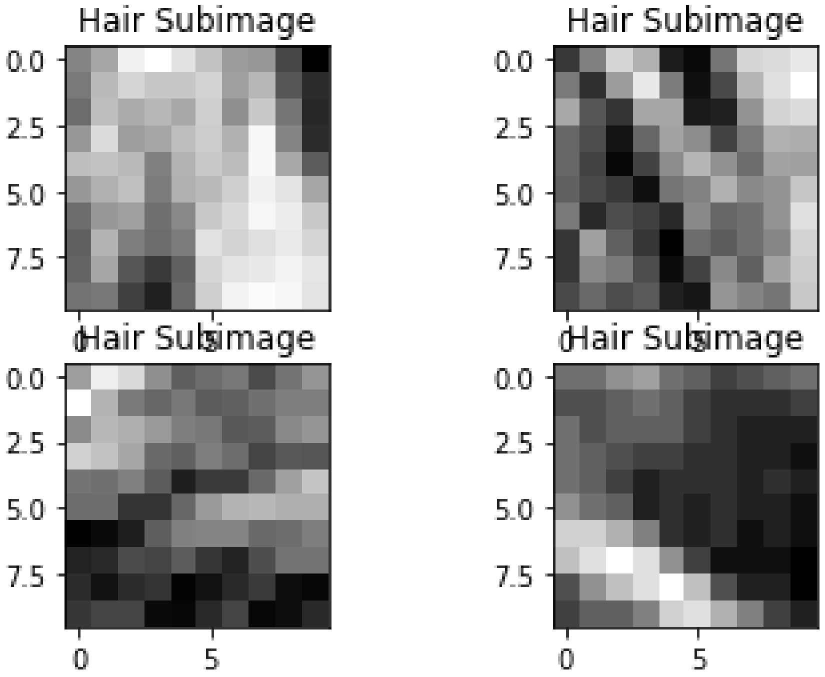
```
4
```

```
constant_image = []
for i in constant_locations:
    constant_image.append(image[i[0]:i[0]+sz,i[1]:i[1]+sz])
```

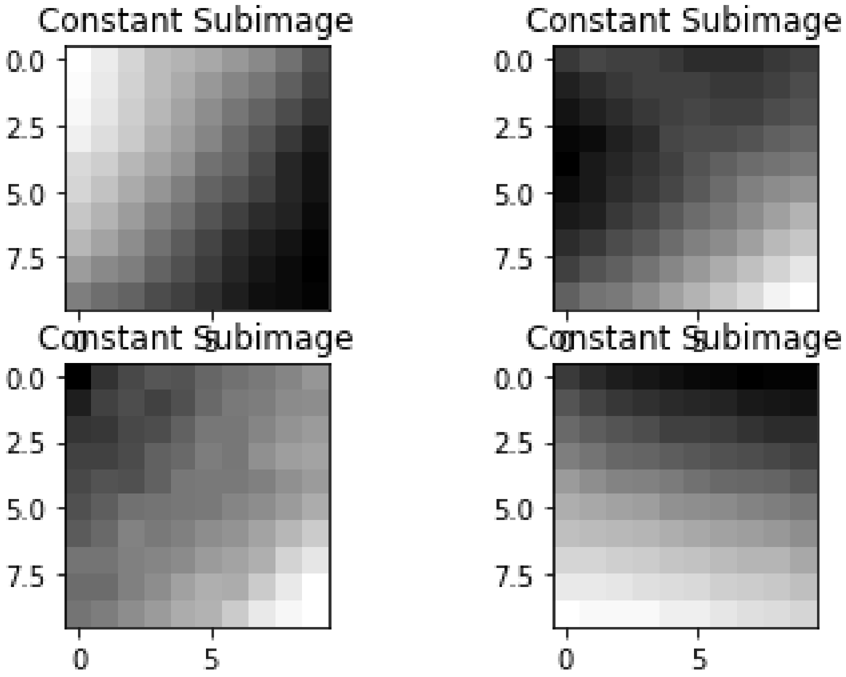
```
len(constant_image)
```

```
4
```

```
for i, subimage in enumerate(hair_image):
    plt.subplot(2,2,i+1)
    plt.imshow(subimage, cmap='gray')
    plt.title('Hair Subimage')
```



```
for i, subimage in enumerate(constant_image):
    plt.subplot(2,2,i+1)
    plt.imshow(subimage, cmap='gray')
    plt.title('Constant Subimage')
```



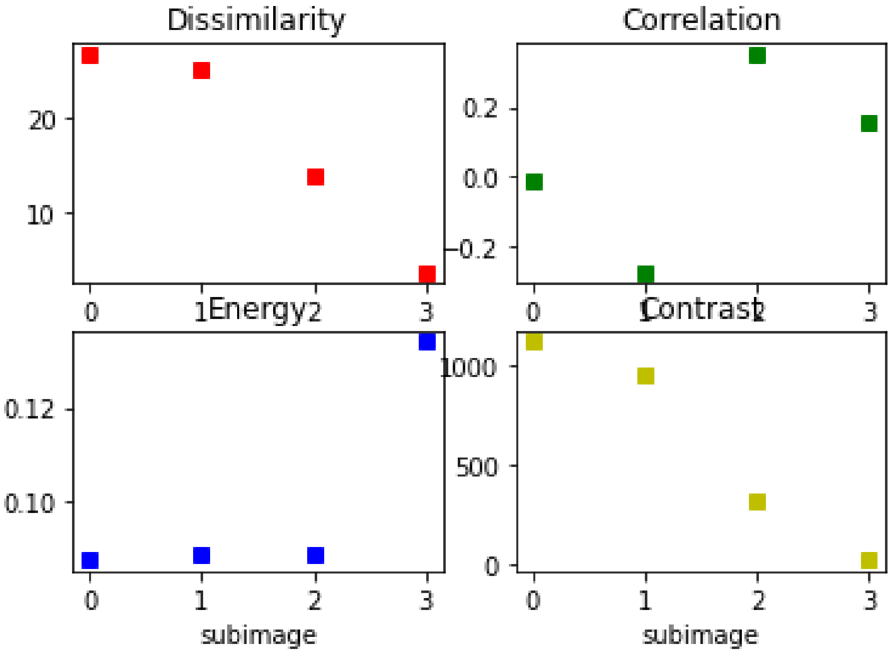
```
diss = []
corr = []
ener = []
cont = []

for i in hair_image:
    glcm = graycomatrix(i, distances = [3], angles = [0],
                        levels = 256, symmetric = True, normed = True)
    diss.append(graycoprops(glcm, 'dissimilarity')[0,0])
    corr.append(graycoprops(glcm, 'correlation')[0,0])
    ener.append(graycoprops(glcm, 'energy')[0,0])
    cont.append(graycoprops(glcm, 'contrast')[0,0])
```

```
len(diss)
```

```
4
```

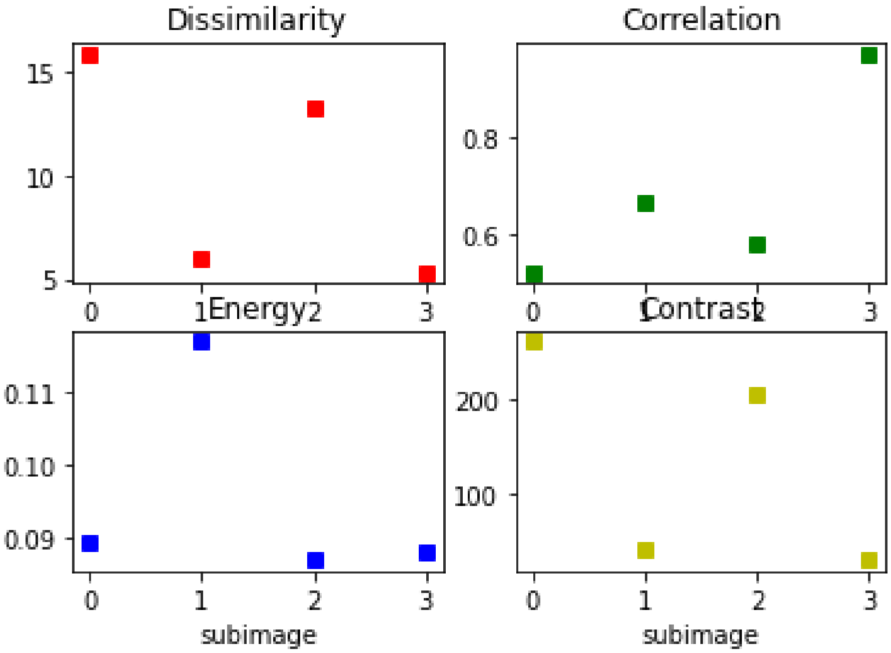
```
diss1 = []
corr1 = []
ener1 = []
cont1 = []
for i in constant_image:
    glcm = graycomatrix(i, distances = [3], angles = [0],
                       levels = 256, symmetric = True, normed = True)
    diss1.append(graycoprops(glcm, 'dissimilarity')[0,0])
    corr1.append(graycoprops(glcm, 'correlation')[0,0])
    ener1.append(graycoprops(glcm, 'energy')[0,0])
    cont1.append(graycoprops(glcm, 'contrast')[0,0])

plt.subplot(2,2,1)
plt.plot(diss,'rs')
plt.title('Dissimilarity')
plt.xlabel('subimage')

plt.subplot(2,2,2)
plt.plot(corr,'gs')
plt.title('Correlation')
plt.xlabel('subimage')

plt.subplot(2,2,3)
plt.plot(ener,'bs')
plt.title('Energy')
plt.xlabel('subimage')

plt.subplot(2,2,4)
plt.plot(cont,'ys')
plt.title('Contrast')
plt.xlabel('subimage')
```

```
Text(0.5, 0, 'subimage')
```



```
# For Constant Image
plt.subplot(2,2,1)
plt.plot(diss1,'rs')
plt.title('Dissimilarity')
plt.xlabel('subimage')

plt.subplot(2,2,2)
plt.plot(corr1,'gs')
plt.title('Correlation')
plt.xlabel('subimage')

plt.subplot(2,2,3)
plt.plot(ener1,'bs')
plt.title('Energy')
plt.xlabel('subimage')

plt.subplot(2,2,4)
plt.plot(cont1,'ys')
plt.title('Contrast')
plt.xlabel('subimage')
```

```
Text(0.5, 0, 'subimage')
```


Dissimilarity
15
10
5
0
1
2
3
Correlation
0.8
0.6
0
1
2
3
Energy
0.11
0.10
0.09
0
1
2
3
subimage
Contrast
200
100
0
1
2
3
subimage