```
import numpy as np
import matplotlib.pyplot as plt
import math
import pandas
import cv2
import seaborn as sb
```

```
image1=plt.imread('image1.jpg')
plt.imshow(image1)
```

```
<matplotlib.image.AxesImage at 0x7f335547a730>
```

```
r1=image1[:,:,0]
g1=image1[:,:,1]
b1=image1[:,:,2]
```

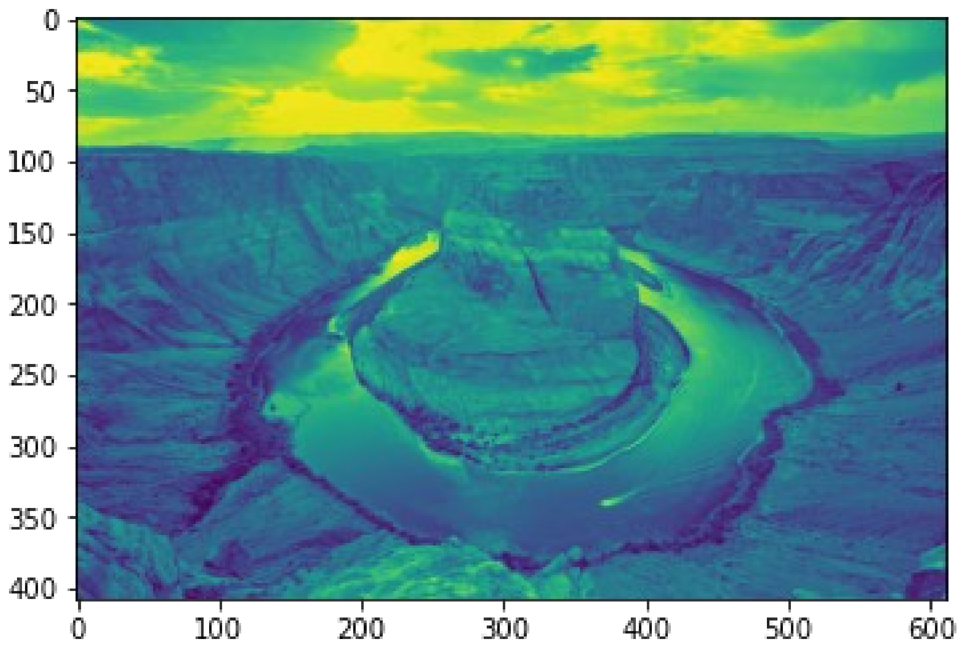
Here I parsed the RGB matrix based on the color

```
image1mat=np.cov(r)
```

```
image1bw = cv2.cvtColor(image1, cv2.COLOR_BGR2GRAY)
```

```
plt.imshow(image1bw)
```

```
<matplotlib.image.AxesImage at 0x7f3355437460>
```



```
print(image1bw)
```

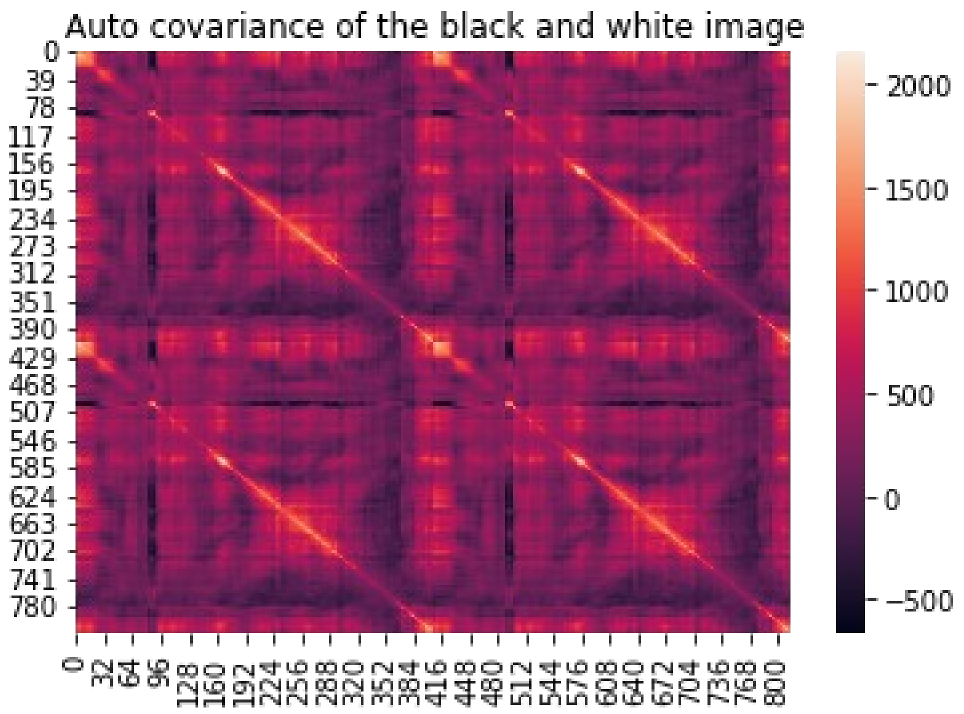
```
[[132 131 130 ... 143 142 141]
 [175 162 149 ... 143 142 141]
 [218 211 203 ... 143 142 141]
 ...
 [ 16  35  52 ... 104 124 115]
 [ 43  47  46 ... 116 127 123]
 [ 81  64  37 ... 116 118 124]]
```

```
autocov1=np.cov(image1bw,image1bw)
```

Here below we can see the different autocrcovariance and the cross covariance heatmaps

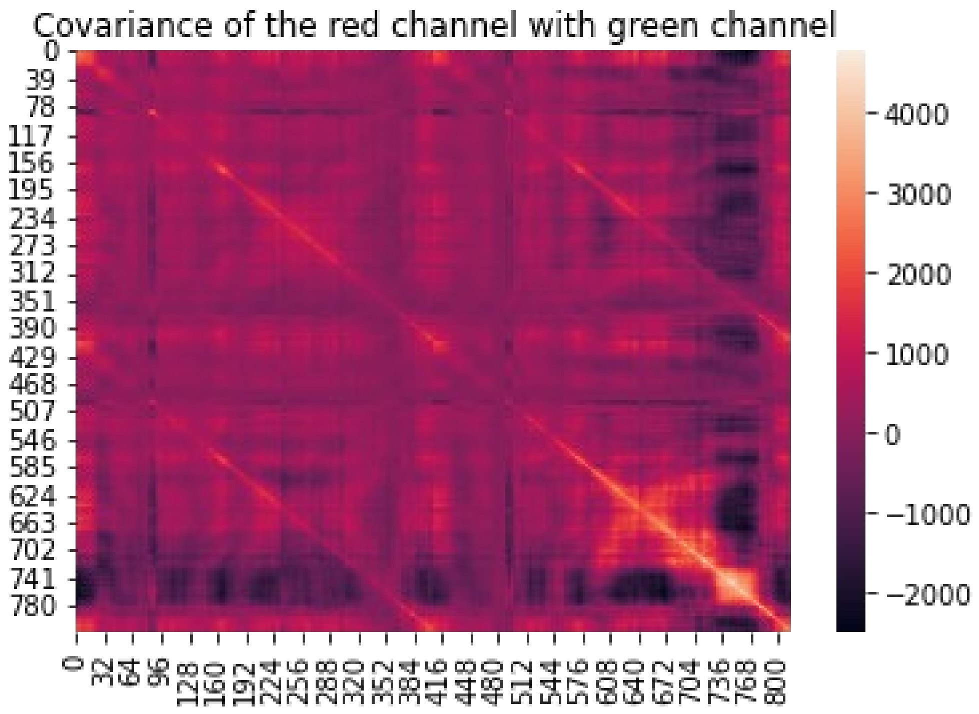
```
sb.heatmap(autocov1)
plt.title('Auto covariance of the black and white image')
```

```
Text(0.5, 1.0, 'Auto covariance of the black and white image')
```



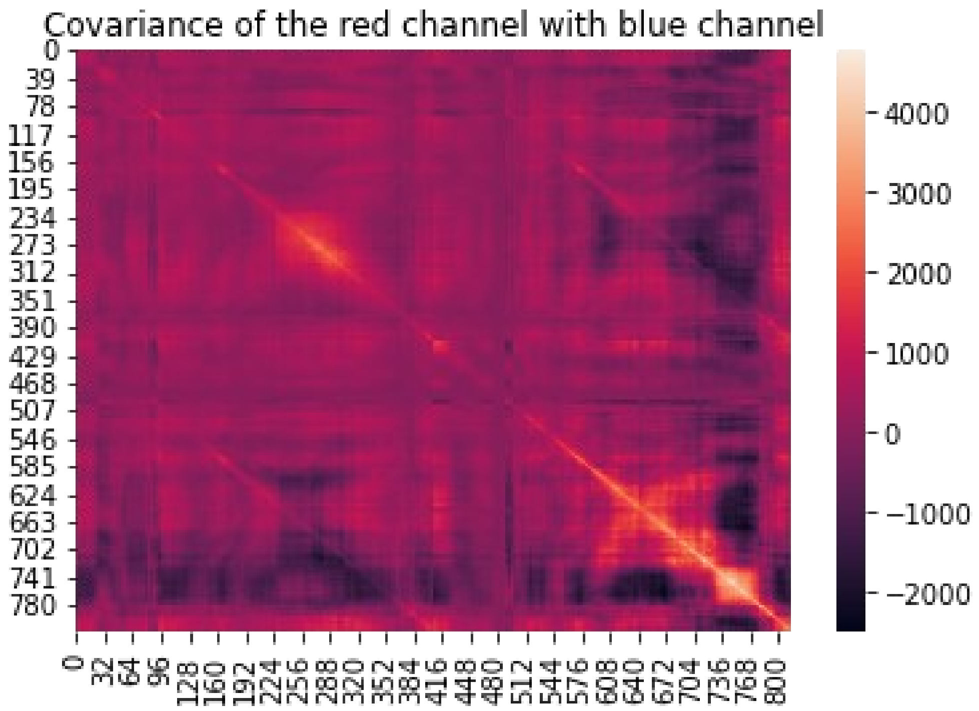
```
crosscov10=np.cov(g1,r1)
sb.heatmap(crosscov10)
plt.title('Covariance of the red channel with green channel')
```

```
Text(0.5, 1.0, 'Covariance of the red channel with green channel')
```



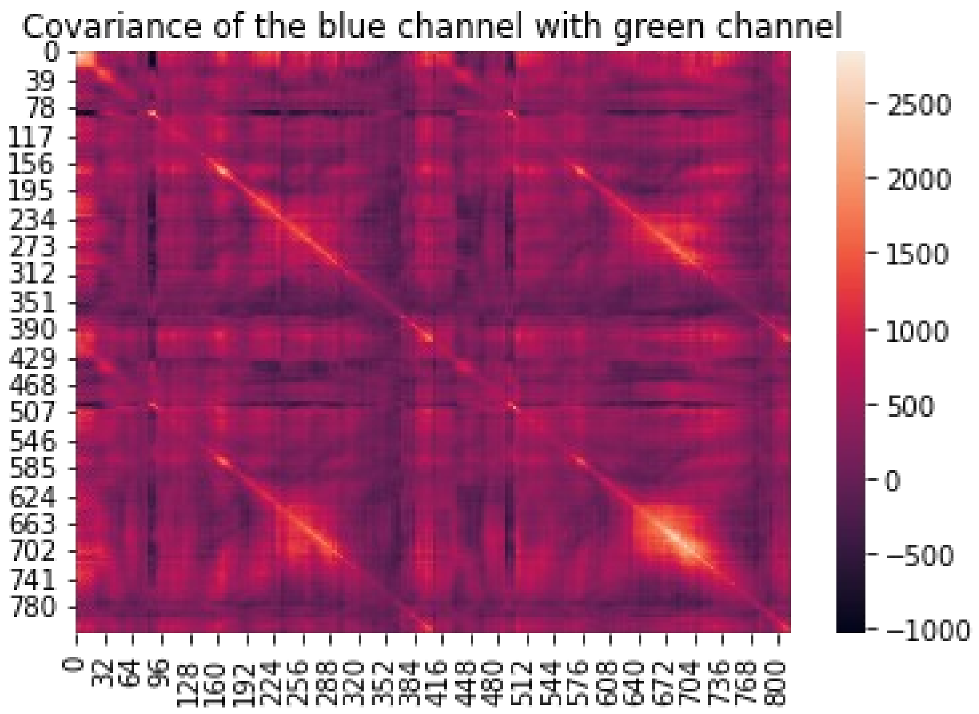
```
crosscov20=np.cov(b1,r1)
sb.heatmap(crosscov20)
plt.title('Covariance of the red channel with blue channel')
```

```
Text(0.5, 1.0, 'Covariance of the red channel with blue channel')
```



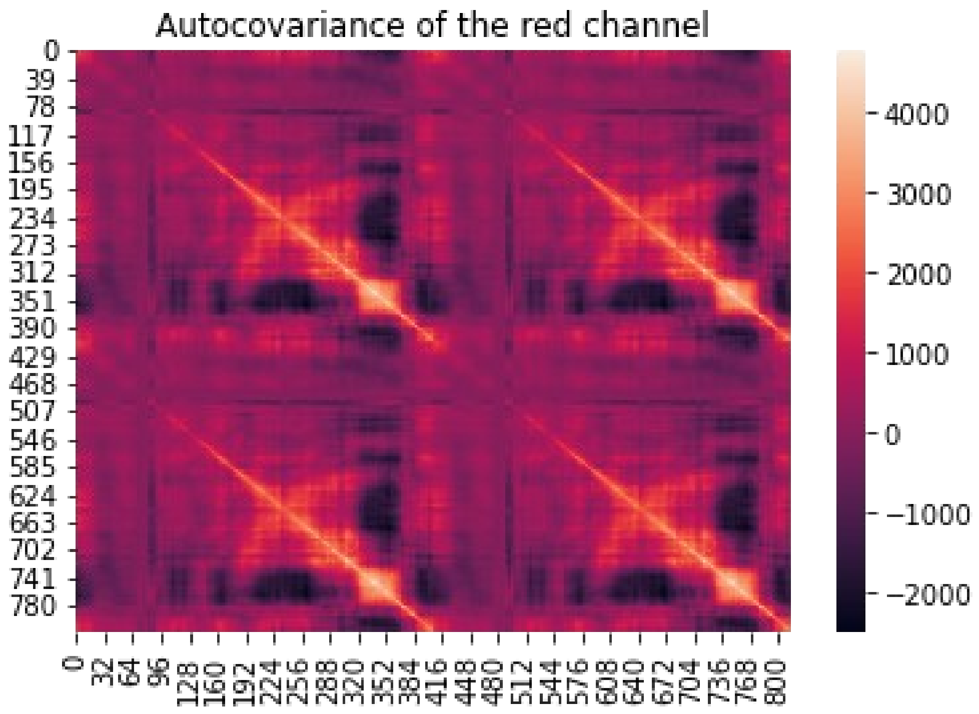
```
crosscov12=np.cov(g1,b1)
sb.heatmap(crosscov12)
plt.title('Covariance of the blue channel with green channel')
```

```
Text(0.5, 1.0, 'Covariance of the blue channel with green channel')
```



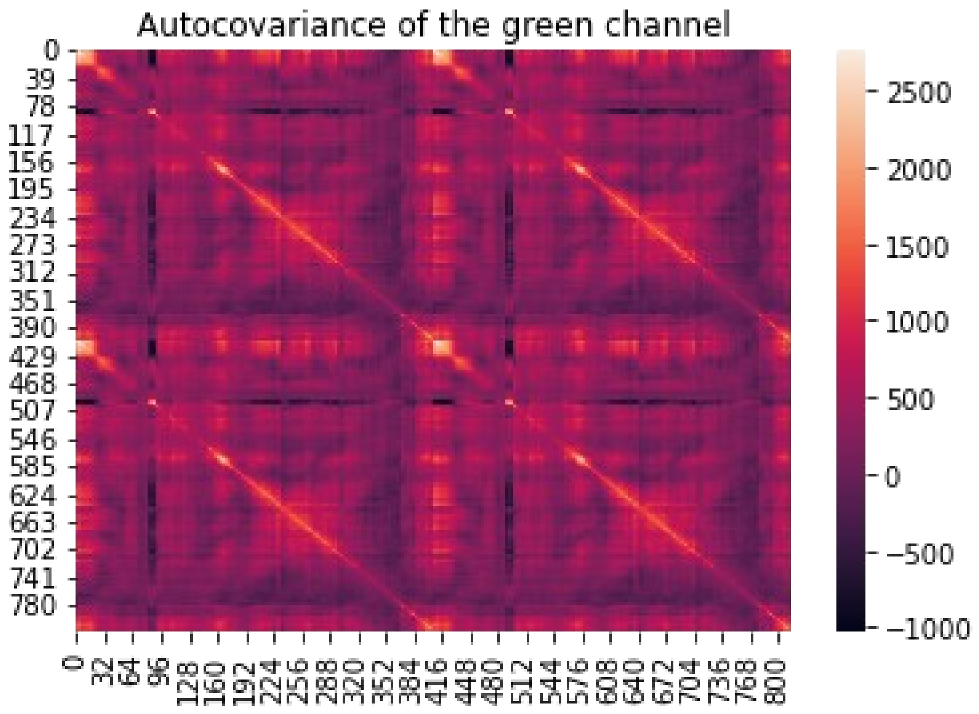
```
crosscov00=np.cov(r1,r1)
sb.heatmap(crosscov00)
plt.title('Autocovariance of the red channel')
```

```
Text(0.5, 1.0, 'Autocovariance of the red channel')
```



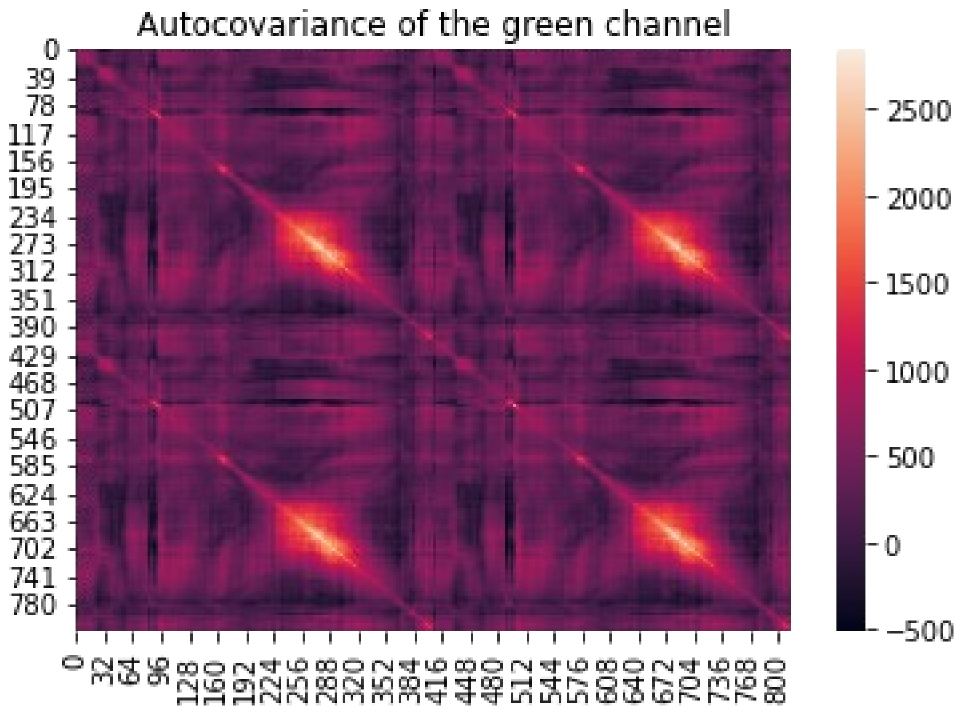
```
crosscov11=np.cov(g1,g1)
sb.heatmap(crosscov11)
plt.title('Autocovariance of the green channel')
```

```
Text(0.5, 1.0, 'Autocovariance of the green channel')
```



```
crosscov22=np.cov(b1,b1)
sb.heatmap(crosscov22)
plt.title('Autocovariance of the green channel')
```

```
Text(0.5, 1.0, 'Autocovariance of the green channel')
```



```
print("The autocovariance of the black and white image is
",np.cov(image1bw,image1bw).mean())
print("The autocovariance of the red channels is
",np.cov(r1,r1).mean())
print("The autocovariance of the green channels is
",np.cov(g1,g1).mean())
print("The autocovariance of the blue channels
is",np.cov(b1,b1).mean())
```

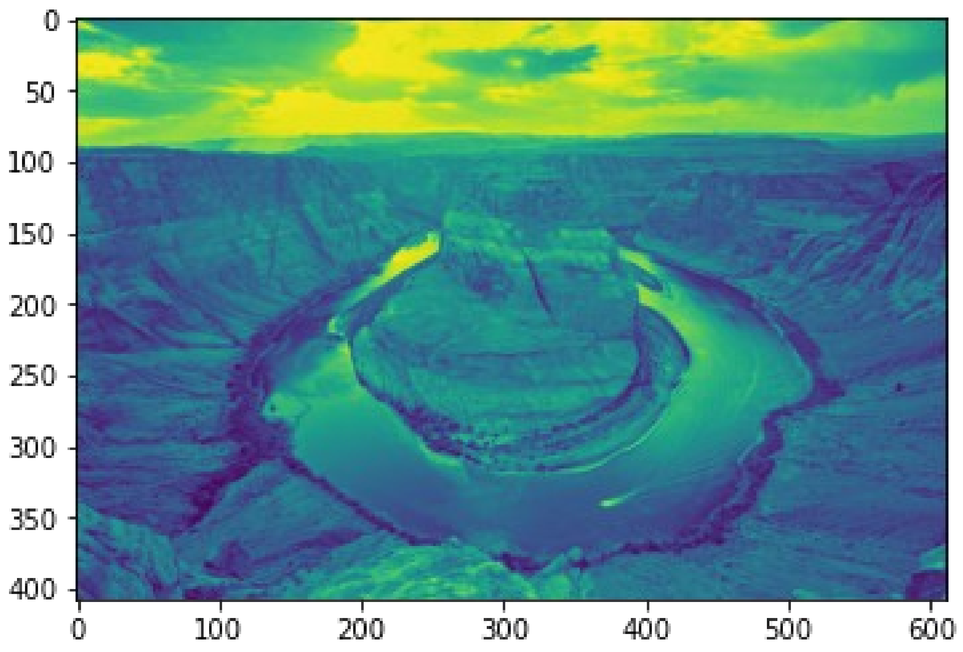
```
The autocovariance of the black and white image is  295.7422573423977
The autocovariance of the red channels is  208.13132394991158
The autocovariance of the green channels is  346.31741293647144
The autocovariance of the blue channels is 371.88001343024206
```

```
#With a shift of 200
newImage=np.roll(image1,200)
newImage=np.transpose(newImage)
newImage=np.roll(newImage,200)

r2=newImage[:,:,0]
g2=newImage[:,:,1]
b2=newImage[:,:,2]
```

```
newImagebw=cv2.cvtColor(image1, cv2.COLOR_BGR2GRAY)
```

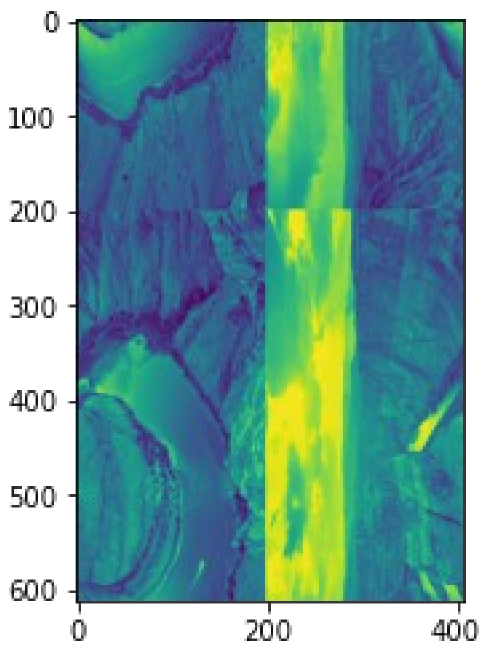
```
plt.imshow(newImagebw)
```

```
<matplotlib.image.AxesImage at 0x7f33565920d0>
```



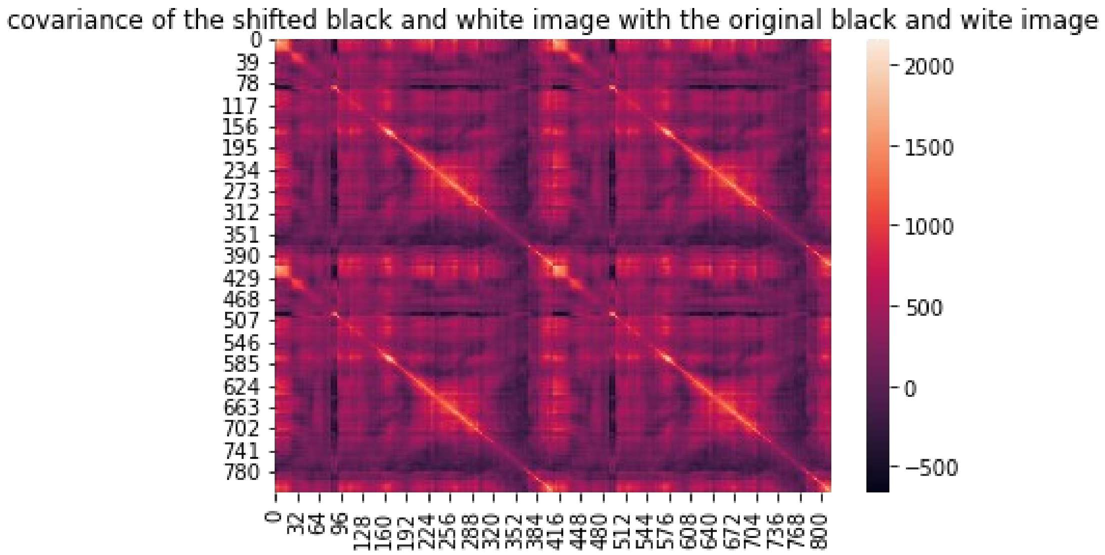
```
#With a shift of 200 in both the directions
newImage=np.roll(newImagebw,200)
newImage=np.transpose(newImage)
newImage=np.roll(newImage,200)
plt.imshow(newImage)
```

```
<matplotlib.image.AxesImage at 0x7f3356572fd0>
```



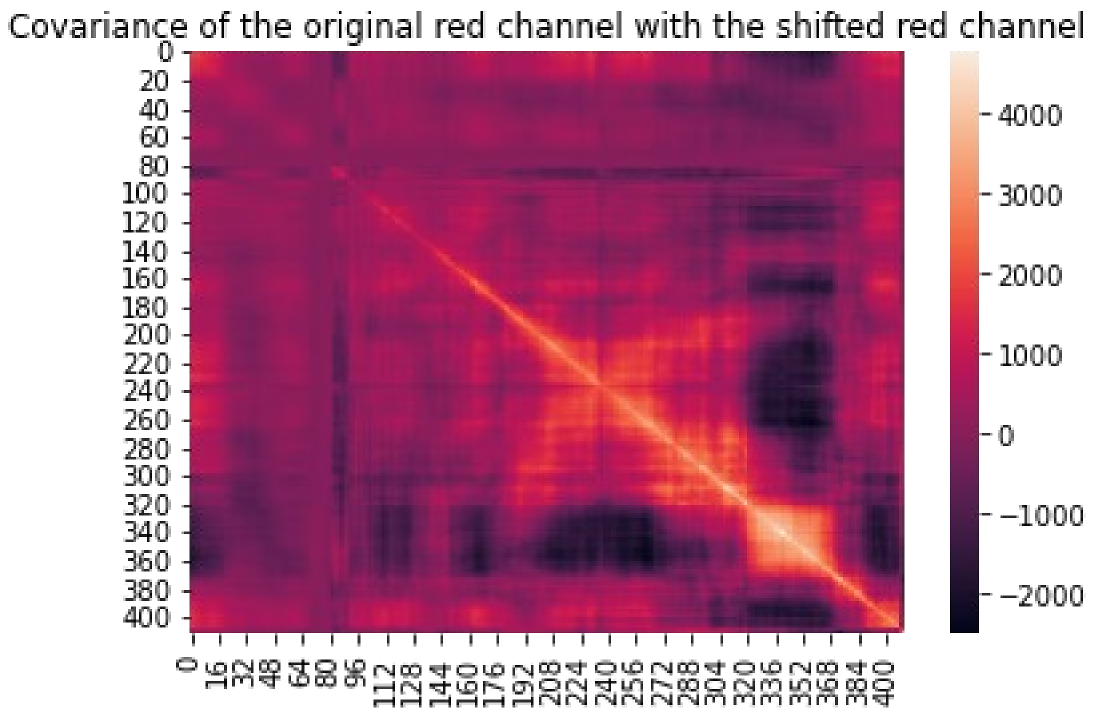
```
sb.heatmap(np.cov(image1bw,newImagebw))
plt.title('covariance of the shifted black and white image with the
original black and wite image')
```

```
Text(0.5, 1.0, 'covariance of the shifted black and white image with
the original black and wite image')
```



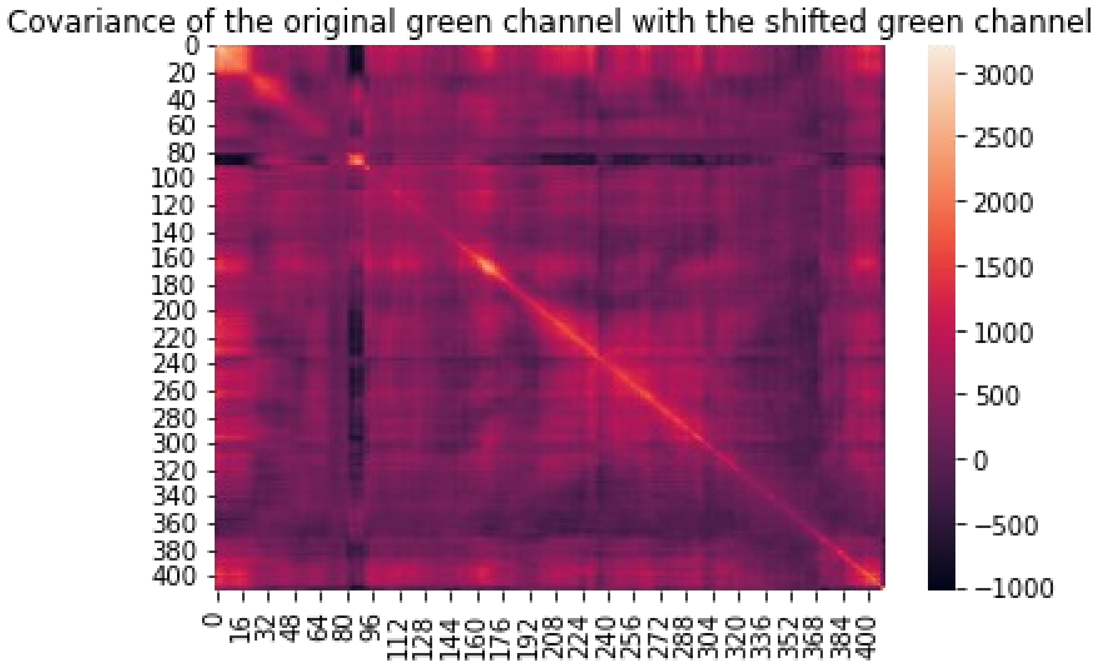
```
sb.heatmap(np.cov(r1,r2))
plt.title('Covariance of the original red channel with the shifted red
channel')
```

```
Text(0.5, 1.0, 'Covariance of the original red channel with the
shifted red channel')
```



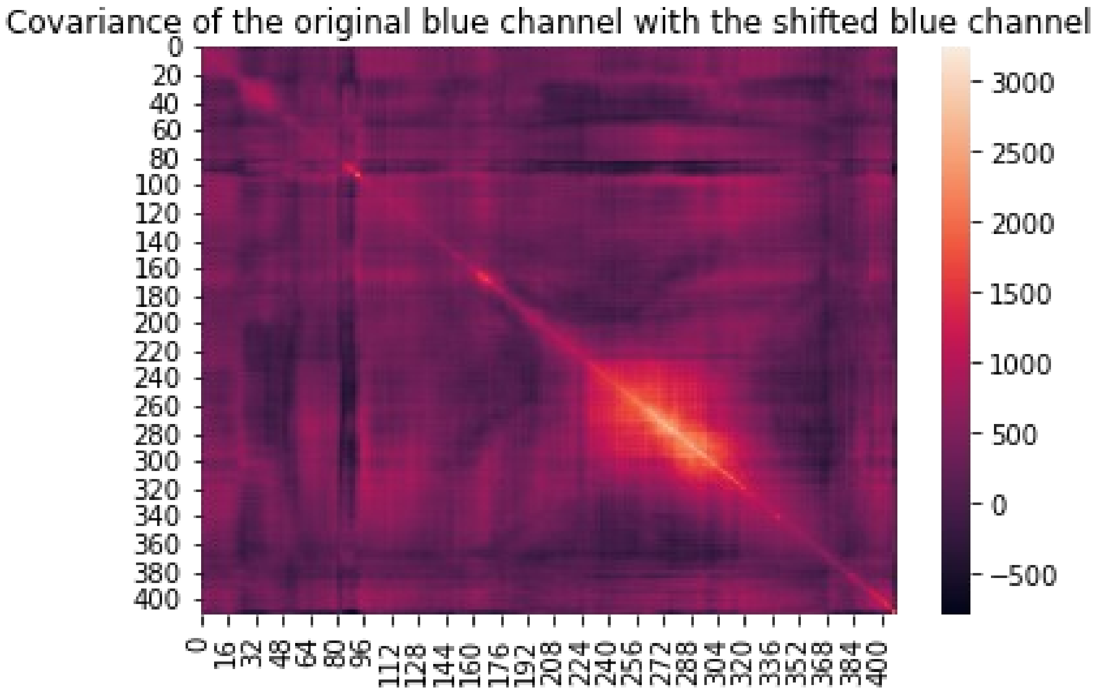
```
sb.heatmap(np.cov(g1,g2))
plt.title('Covariance of the original green channel with the shifted
green channel')
```

```
Text(0.5, 1.0, 'Covariance of the original green channel with the
shifted green channel')
```



```
sb.heatmap(np.cov(b1,b2))
plt.title('Covariance of the original blue channel with the shifted
blue channel')
```

```
Text(0.5, 1.0, 'Covariance of the original blue channel with the
shifted blue channel')
```



covariance with a cropped version of the image

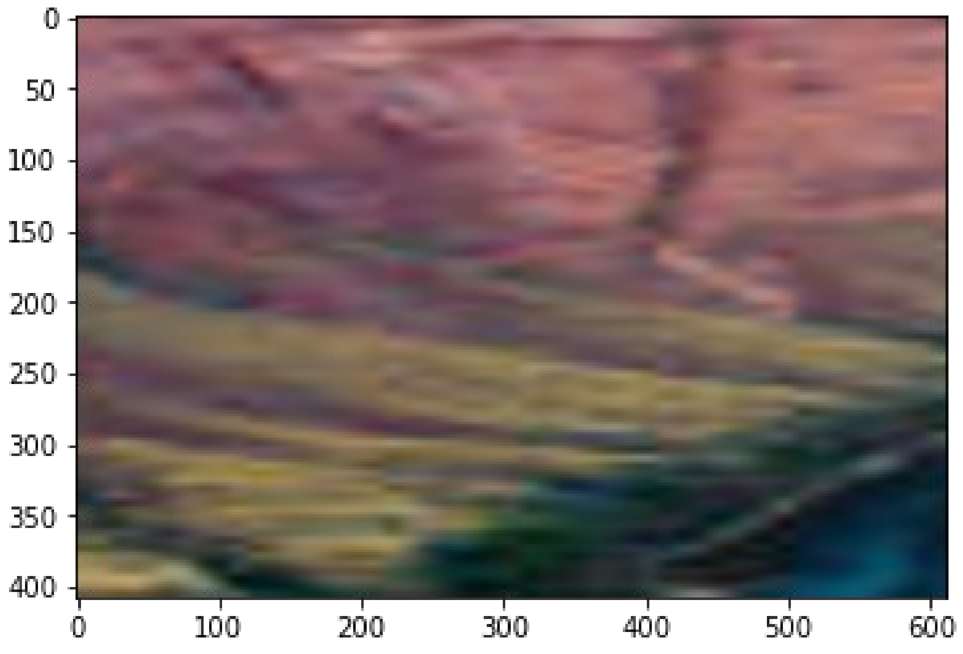
```
#ci=plt.imread('croppedimage1.png')
image1.shape
```

```
(408, 612, 3)
```

```
#plt.imshow(ci)

# #resizing the image
# shape=image1bw.shape
# print(shape)

# cibw=cv2.cvtColor(ci, cv2.COLOR_BGR2GRAY)

# plt.imshow(cibw)
# print(cibw.shape)
# cibw=cibw.resize(shape)
```

```
from PIL import Image

# Opens a image in RGB mode
```

```
im = Image.open(r"image1.jpg")

# Size of the image in pixels (size of original image)
# (This is not mandatory)
width, height = im.size

# Setting the points for cropped image
left = length/4
top = breadth / 4
right = 1.5* length/4
bottom = 1.5* breadth / 4

# Cropped image of above dimension
# (It will not change original image)
im1 = im.crop((left, top, right, bottom))
newsize = (612,408)
im1 = im1.resize(newsize)
# Shows the image in image viewer
ci=plt.imshow(im1)
```



```
cibw=cv2.cvtColor(im1, cv2.COLOR_BGR2GRAY)

----------------------------------------------------------------------
-----
error                                     Traceback (most recent call
last)
Input In [331], in <cell line: 1>()
----> 1 cibw=cv2.cvtColor(im1, cv2.COLOR_BGR2GRAY)

error: OpenCV(4.6.0) :-1: error: (-5:Bad argument) in function
```

```
'cvtColor'
> Overload resolution failed:
>  - src is not a numpy array, neither a scalar
>  - Expected Ptr<cv::UMat> for argument 'src'
```

```
sb.heatmap(np.cov(image1bw,cibw))
plt.title('Covariance of the original blue channel with the shifted
blue channel')
```

```
-----------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
Input In [332], in <cell line: 1>()
----> 1 sb.heatmap(np.cov(image1bw,cibw))
      2 plt.title('Covariance of the original blue channel with the
shifted blue channel')

File <__array_function__ internals>:5, in cov(*args, **kwargs)

File
~/anaconda3/lib/python3.9/site-packages/numpy/lib/function_base.py:247
7, in cov(m, y, rowvar, bias, ddof, fweights, aweights, dtype)
   2475     if not rowvar and y.shape[0] != 1:
   2476         y = y.T
-> 2477     X = np.concatenate((X, y), axis=0)
   2479 if ddof is None:
   2480     if bias == 0:

File <__array_function__ internals>:5, in concatenate(*args, **kwargs)

ValueError: all the input array dimensions for the concatenation axis
must match exactly, but along dimension 1, the array at index 0 has
size 612 and the array at index 1 has size 156
```