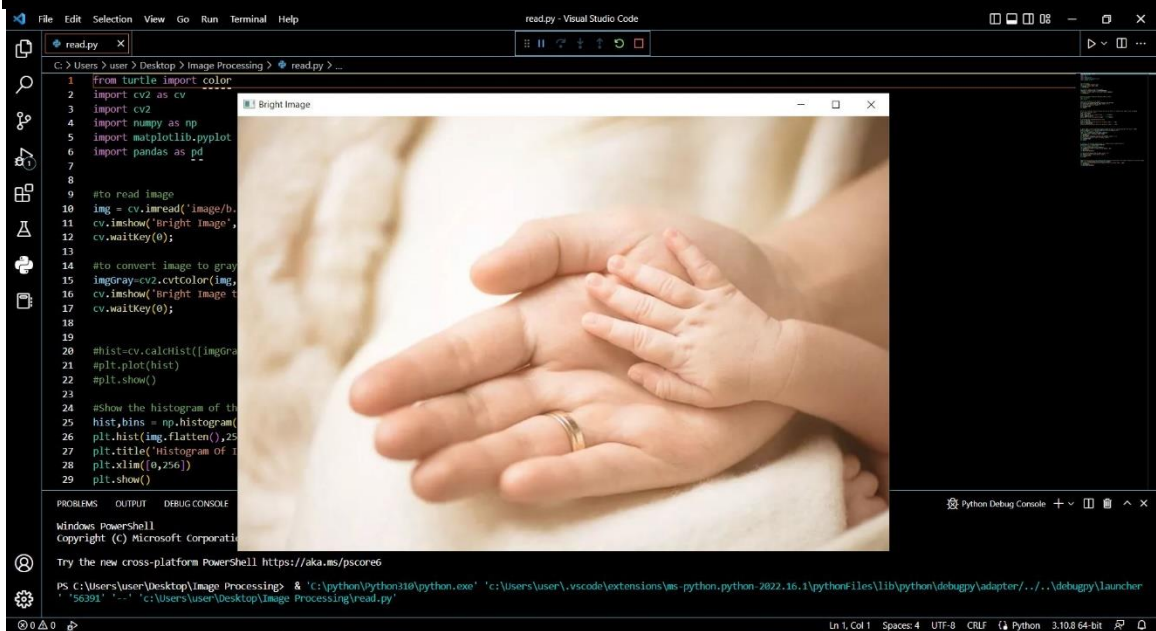


```
from turtle import color
import cv2 as cv
import cv2
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

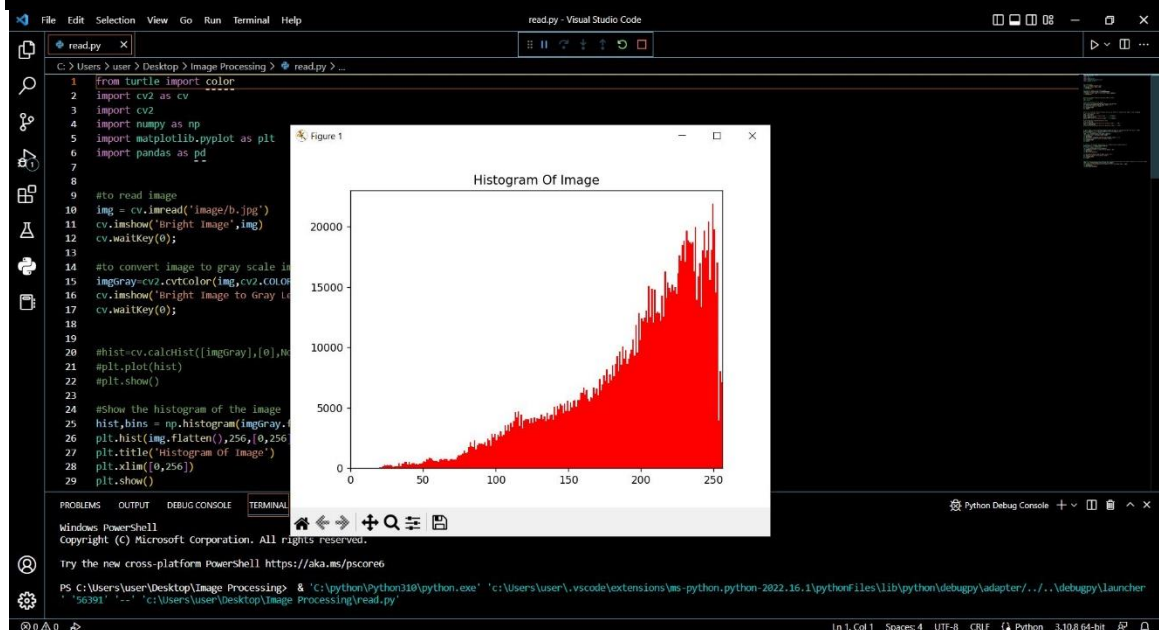
#to read image
img = cv.imread('image/b.jpg')
cv.imshow('Bright Image',img)
cv.waitKey(0);
```



```
#to convert image to gray scale image
imgGray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
cv.imshow('Bright Image to Gray Level Image',imgGray)
cv.waitKey(0);
```



```
#Show the histogram of the image
hist,bins = np.histogram(imgGray.flatten(),256,[0,256])
plt.hist(img.flatten(),256,[0,256], color = 'r')
plt.title('Histogram Of Image')
plt.xlim([0,256])
plt.show()
```



```
#Print the necessary features/values that can be useful to identify the
shape of the histogram
# Calculate the mean value
mean = imgGray.mean()
print("Mean Value for grayscale image : " + str(mean))

# Calculate the standard deviation value
std= np.std(imgGray)
print("Standard Deviation Value for grayscale image : " , std)
```

```
29
30
31 #Print the necessary features/values that can be useful to identify the shape of the histogram
32 # Calculate the mean value
33 mean = imgGray.mean()
34 print("Mean Value for grayscale image : " + str(mean))
35
36 # calculate the standard deviation value
37 std= np.std(imgGray)
38 print("Standard Deviation Value for grayscale image :", std)
39
40
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** JUPYTER

Python Debug Console + - [] ^ x

Try the new cross-platform PowerShell <https://aka.ms/powershell>

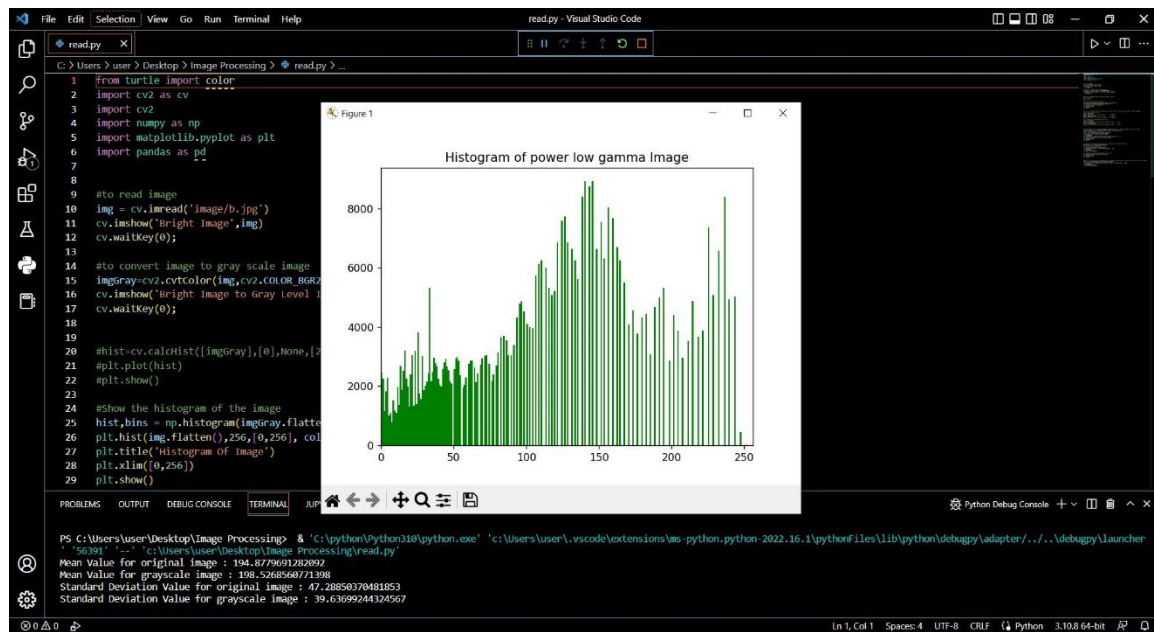
```
PS C:\Users\User\Desktop\Image Processing> & 'C:\python\Python310\python.exe' 'c:\Users\User\.vscode\extensions\ms-python.python-2022.16.1\python\lib\python\debugpy\adapter/.../debugpy_launcher.py' -c:\Users\User\Desktop\Image Processing\read.py
Mean Value for grayscale image : 108.5268560771398
Standard Deviation Value for grayscale image : 39.6369924432567
PS C:\Users\User\Desktop\Image Processing>
```

Ln 56, Col 49 Spaces: 4 UTF-8 CRLF Python 3.10.8 64-bit

```

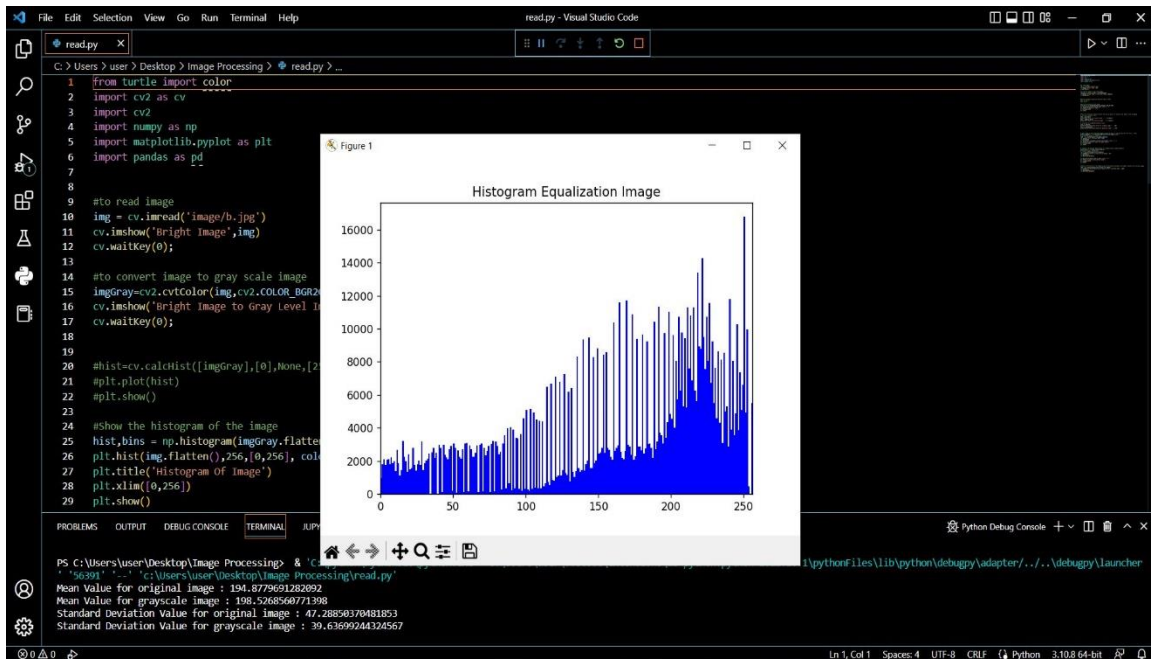
# Apply Gamma on the normalised image and then multiply by scaling
constant (For 8 bit, c=255)
power_low_gamma = np.array(255*(imgGray/255)**3.9,dtype='uint8')
# Display the images in subplots
img3 = cv2.hconcat([power_low_gamma,imgGray])
cv2.imshow('Power Low Gamma Image',img3)
cv2.waitKey(0)
plt.hist(power_low_gamma.flatten(),256,[0,256], color = 'g')
plt.title('Histogram of power low gamma Image')
plt.xlim([0,256])
plt.show()

```



```
# creating a Histograms Equalization of a image using cv2.equalizeHist()
equalization = cv2.equalizeHist(imgGray)
# stacking images side-by-side
res = np.hstack((equalization,imgGray))
cv2.imshow('Histograms Equalization Image', res)
cv2.waitKey(0)
cv2.destroyAllWindows()

plt.hist(res.flatten(),256,[0,256], color = 'b')
plt.title('Histogram Equalization Image')
plt.xlim([0,256])
plt.show()
```




```
#Enhance the contrast of the image using the following techniques and
compare between the resulting images
img4 = cv2.hconcat([equalization,power_low_gamma])
cv2.imshow('Histogram Equalization Image Vs Power Low Gamma Image ',img4)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

