



# RELATÓRIO TÉCNICO

Wesley de Oliveira Mendes - 828.507

## Tarefa 01 - Operações Pontuais

Prof. Rodrigo de Oliveira Plotze

# Processamento de Imagens e Imagens

Engenharia da Computação - 2021.01

**Wesley de Oliveira Mendes, 828.507**

## Tarefa 01 - Operações Pontuais

- Objetivo
  - Aplicar técnicas fundamentais de processamento de imagens.

### Download das imagens

```
In [24]: !wget 'https://homepages.cae.wisc.edu/~ece533/images/pool.png'
!wget 'https://homepages.cae.wisc.edu/~ece533/images/monarch.png'
!wget 'https://homepages.cae.wisc.edu/~ece533/images/girl.png'
!wget 'https://homepages.cae.wisc.edu/~ece533/images/sails.png'
!wget 'https://homepages.cae.wisc.edu/~ece533/images/tulips.png'
!wget 'https://homepages.cae.wisc.edu/~ece533/images/frymire.png'
!wget 'https://homepages.cae.wisc.edu/~ece533/images/watch.png'

--2021-04-04 21:42:11-- https://homepages.cae.wisc.edu/~ece533/images/pool.png
Resolving homepages.cae.wisc.edu (homepages.cae.wisc.edu)... 144.92.13.84, 2607:f388:1082:0:279f:7b00:a091:17d7
Connecting to homepages.cae.wisc.edu (homepages.cae.wisc.edu)|144.92.13.84|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 186897 (183K) [image/png]
Saving to: 'pool.png.1'

pool.png.1      100%[=====] 182.52K   327KB/s    in 0.6s

2021-04-04 21:42:12 (327 KB/s) - 'pool.png.1' saved [186897/186897]

--2021-04-04 21:42:12-- https://homepages.cae.wisc.edu/~ece533/images/monarch.png
Resolving homepages.cae.wisc.edu (homepages.cae.wisc.edu)... 144.92.13.84, 2607:f388:1082:0:279f:7b00:a091:17d7
Connecting to homepages.cae.wisc.edu (homepages.cae.wisc.edu)|144.92.13.84|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 614179 (600K) [image/png]
Saving to: 'monarch.png.1'

monarch.png.1     100%[=====] 599.78K   659KB/s    in 0.9s

2021-04-04 21:42:14 (659 KB/s) - 'monarch.png.1' saved [614179/614179]

--2021-04-04 21:42:14-- https://homepages.cae.wisc.edu/~ece533/images/girl.png
Resolving homepages.cae.wisc.edu (homepages.cae.wisc.edu)... 144.92.13.84, 2607:f388:1082:0:279f:7b00:a091:17d7
Connecting to homepages.cae.wisc.edu (homepages.cae.wisc.edu)|144.92.13.84|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 623591 (609K) [image/png]
Saving to: 'girl.png.1'

girl.png.1      100%[=====] 608.98K   670KB/s    in 0.9s

2021-04-04 21:42:16 (670 KB/s) - 'girl.png.1' saved [623591/623591]

--2021-04-04 21:42:16-- https://homepages.cae.wisc.edu/~ece533/images/sails.png
Resolving homepages.cae.wisc.edu (homepages.cae.wisc.edu)... 144.92.13.84, 2607:f388:1082:0:279f:7b00:a091:17d7
Connecting to homepages.cae.wisc.edu (homepages.cae.wisc.edu)|144.92.13.84|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 806850 (788K) [image/png]
Saving to: 'sails.png.1'

sails.png.1      100%[=====] 787.94K   848KB/s    in 0.9s

2021-04-04 21:42:18 (848 KB/s) - 'sails.png.1' saved [806850/806850]

--2021-04-04 21:42:18-- https://homepages.cae.wisc.edu/~ece533/images/tulips.png
Resolving homepages.cae.wisc.edu (homepages.cae.wisc.edu)... 144.92.13.84, 2607:f388:1082:0:279f:7b00:a091:17d7
Connecting to homepages.cae.wisc.edu (homepages.cae.wisc.edu)|144.92.13.84|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 679233 (663K) [image/png]
Saving to: 'tulips.png.1'

tulips.png.1     100%[=====] 663.31K   727KB/s    in 0.9s

2021-04-04 21:42:20 (727 KB/s) - 'tulips.png.1' saved [679233/679233]

--2021-04-04 21:42:20-- https://homepages.cae.wisc.edu/~ece533/images/frymire.png
Resolving homepages.cae.wisc.edu (homepages.cae.wisc.edu)... 144.92.13.84, 2607:f388:1082:0:279f:7b00:a091:17d7
Connecting to homepages.cae.wisc.edu (homepages.cae.wisc.edu)|144.92.13.84|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 252124 (246K) [image/png]
Saving to: 'frymire.png.1'

frymire.png.1     100%[=====] 246.21K   337KB/s    in 0.7s

2021-04-04 21:42:21 (337 KB/s) - 'frymire.png.1' saved [252124/252124]

--2021-04-04 21:42:21-- https://homepages.cae.wisc.edu/~ece533/images/watch.png
Resolving homepages.cae.wisc.edu (homepages.cae.wisc.edu)... 144.92.13.84, 2607:f388:1082:0:279f:7b00:a091:17d7
Connecting to homepages.cae.wisc.edu (homepages.cae.wisc.edu)|144.92.13.84|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 697056 (681K) [image/png]
Saving to: 'watch.png.1'

watch.png.1      100%[=====] 680.72K   735KB/s    in 0.9s

2021-04-04 21:42:24 (735 KB/s) - 'watch.png.1' saved [697056/697056]
```

## Imports

```
In [25]: import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
```

## Code

### Exercício 1

Realize as seguintes etapas de processamento de imagens:

- Carregue três imagens de sua preferência: img1, img2 e img3.
- Realize a conversão da img2 para tons de cinza.
- Realize a conversão da img3 para preto e branco.
- Apresente na tela a quantidade total de pixels da img1.
- Apresente na tela o número de linhas da img2.
- Apresente na tela o número de colunas da img3.
- Apresente na tela, lado a lado, as três imagens.

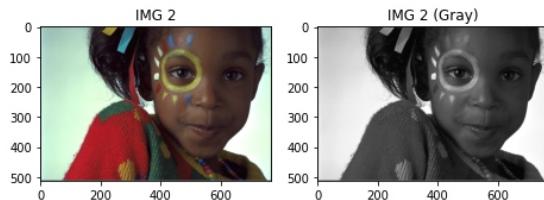
```
In [26]: # Carregue três imagens de sua preferência: img1, img2 e img3.
exel_img1 = cv.imread('pool.png')
exel_img1 = cv.cvtColor(exel_img1, cv.COLOR_BGR2RGB)

exel_img2 = cv.imread('girl.png')
exel_img2 = cv.cvtColor(exel_img2, cv.COLOR_BGR2RGB)

exel_img3 = cv.imread('monarch.png')
exel_img3 = cv.cvtColor(exel_img3, cv.COLOR_BGR2RGB)
```

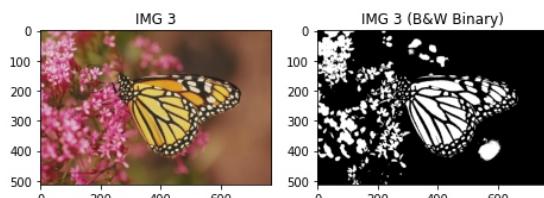
```
In [27]: # Realize a conversão da img2 para tons de cinza.
exel_img2_gray = cv.cvtColor(exel_img2, cv.COLOR_RGB2GRAY)

plt.figure(figsize=(16, 8))
plt.subplot(141), plt.imshow(exel_img2), plt.title('IMG 2')
plt.subplot(142), plt.imshow(exel_img2_gray, cmap='gray'), plt.title('IMG 2 (Gray)')
plt.show()
```



```
In [28]: # Realize a conversão da img3 para preto e branco.
exel_img3_gray = cv.cvtColor(exel_img3, cv.COLOR_RGB2GRAY)
_, exel_img3_bwb = cv.threshold(exel_img3_gray, 127, 255, cv.THRESH_BINARY)

plt.figure(figsize=(16, 8))
plt.subplot(141), plt.imshow(exel_img3), plt.title('IMG 3')
plt.subplot(142), plt.imshow(exel_img3_bwb, cmap='gray'), plt.title('IMG 3 (B&W Binary)')
plt.show()
```



```
In [29]: # Apresente na tela a quantidade total de pixels da img1.
print(f'Total pixels = {exel_img1.size}')

Total pixels = 585990
```

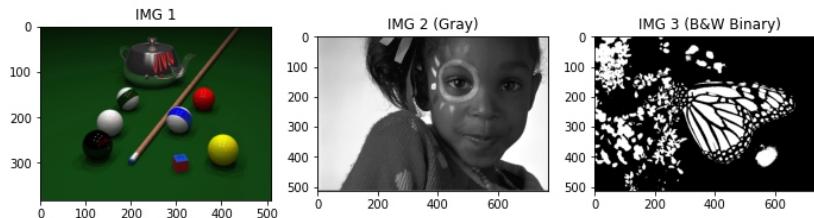
```
In [30]: # Apresente na tela o número de linhas da img2.
print(f'Número de linhas = {exel_img2.shape[0]}')

Número de linhas = 512
```

```
In [31]: # Apresente na tela o número de colunas da img3.
print(f'Número de colunas = {exel_img3.shape[1]}')

Número de colunas = 768
```

```
In [32]: # Apresente na tela, lado a lado, as três imagens.
plt.figure(figsize=(16, 8))
plt.subplot(141), plt.imshow(exel_img1), plt.title('IMG 1')
plt.subplot(142), plt.imshow(exel_img2_gray, cmap='gray'), plt.title('IMG 2 (Gray)')
plt.subplot(143), plt.imshow(exel_img3_bwb, cmap='gray'), plt.title('IMG 3 (B&W Binary)')
plt.show()
```



## Exercício 2

Utilize técnicas de processamento de imagens para:

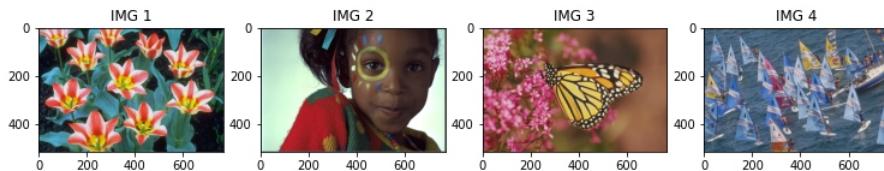
- Carregue quatro imagens de sua preferência: img1, img2, img3 e img4.
- Apresentar na tela todas as imagens carregadas.
- Aplicar a operação pontual para gerar a img5 e apresente o resultado na tela:

```
img5 = (img1 + img2) + (img3 * 0.5) + (img4 - img1)
```

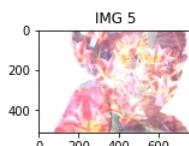
- Criar a img6 a partir da operação pontual de negação da img1. Apresentar comparativamente o histograma da img1 e da img6.

```
In [33]: # Carregue quatro imagens de sua preferência: img1, img2, img3 e img4.  
exe2_img1 = cv.imread('tulips.png')  
exe2_img1 = cv.cvtColor(exe2_img1, cv.COLOR_BGR2RGB)  
  
exe2_img2 = cv.imread('girl.png')  
exe2_img2 = cv.cvtColor(exe2_img2, cv.COLOR_BGR2RGB)  
  
exe2_img3 = cv.imread('monarch.png')  
exe2_img3 = cv.cvtColor(exe2_img3, cv.COLOR_BGR2RGB)  
  
exe2_img4 = cv.imread('sails.png')  
exe2_img4 = cv.cvtColor(exe2_img4, cv.COLOR_BGR2RGB)
```

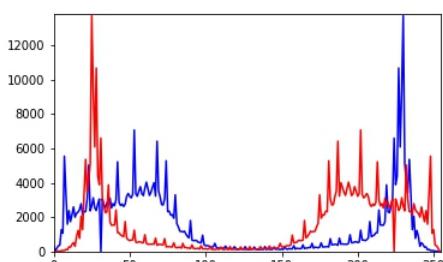
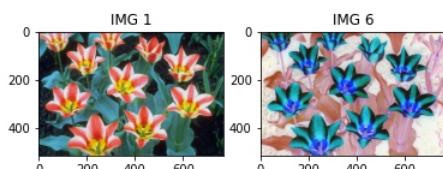
```
In [34]: # Apresentar na tela todas as imagens carregadas.  
plt.figure(figsize=(16, 8))  
plt.subplot(151), plt.imshow(exe2_img1), plt.title('IMG 1')  
plt.subplot(152), plt.imshow(exe2_img2), plt.title('IMG 2')  
plt.subplot(153), plt.imshow(exe2_img3), plt.title('IMG 3')  
plt.subplot(154), plt.imshow(exe2_img4), plt.title('IMG 4')  
plt.show()
```



```
In [35]: # Aplicar a operação pontual para gerar a img5 e apresente o resultado na tela.  
# img5 = (img1 + img2) + (img3 * 0.5) + (img4 - img1)  
img_sum1 = cv.add(exe2_img1, exe2_img2)  
img_mult = cv.multiply(exe2_img3, 0.5)  
img_sum2 = cv.subtract(exe2_img4, exe2_img1)  
  
img_sum3 = cv.add(img_sum1, img_mult)  
exe2_img5 = cv.add(img_sum3, img_sum2)  
  
plt.figure(figsize=(16, 8))  
plt.subplot(161), plt.imshow(exe2_img5), plt.title('IMG 5')  
plt.show()
```



```
In [36]: # Criar a img6 a partir da operação pontual de negação da img1.  
# Apresentar comparativamente o histograma da img1 e da img6.  
exe2_img6 = cv.bitwise_not(exe2_img1)  
  
exe2_hist1 = cv.calcHist([exe2_img1], [0], None, [256], [0, 255])  
exe2_hist6 = cv.calcHist([exe2_img6], [0], None, [256], [0, 255])  
  
plt.figure(figsize=(16, 8))  
plt.subplot(151), plt.imshow(exe2_img1), plt.title('IMG 1')  
plt.subplot(152), plt.imshow(exe2_img6), plt.title('IMG 6')  
plt.show()  
  
plt.figure(figsize=(26.6, 8))  
plt.subplot(242), plt.plot(exe2_hist1, '-', color='blue'), plt.plot(exe2_hist6, '-', color='red'), plt.xlim(0, 255), plt.ylim(0, max(exe2_hist1) + 100)  
plt.show()
```



### Exercício 3

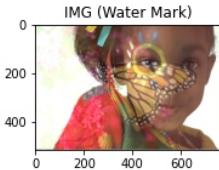
Utilizando operações pontuais demonstre o processo de geração de uma marca d'água a partir do carregamento de duas imagens. Vocês podem escolher imagens coloridas ou tons de cinza.

```
In [37]: exe3_img1 = cv.imread('girl.png')
exe3_img1 = cv.cvtColor(exe3_img1, cv.COLOR_BGR2RGB)

exe3_img2 = cv.imread('monarch.png')
exe3_img2 = cv.cvtColor(exe3_img2, cv.COLOR_BGR2RGB)

water_mark = cv.addWeighted(exe3_img1, 1.0, exe3_img2, 0.5, 0)

plt.figure(figsize=(16, 8))
plt.subplot(151), plt.imshow(water_mark), plt.title('IMG (Water Mark)')
plt.show()
```



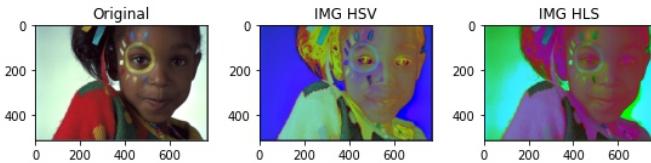
### Exercício 4

Faça uma pesquisa sobre o uso de espaço de cores com a biblioteca OpenCV. Demonstre o processo de conversão de uma imagem RGB em HSV (ou HLS). Ao final, apresente na tela o resultado da imagem convertida.

```
In [38]: exe4_img1 = cv.imread('girl.png')
exe4_img1 = cv.cvtColor(exe4_img1, cv.COLOR_BGR2RGB)

exe4_img_hsv = cv.cvtColor(exe4_img1, cv.COLOR_BGR2HSV)
exe4_img_hls = cv.cvtColor(exe4_img1, cv.COLOR_BGR2HLS)

plt.figure(figsize=(16, 8))
plt.subplot(451), plt.imshow(exe4_img1), plt.title('Original')
plt.subplot(452), plt.imshow(exe4_img_hsv, cmap='hsv'), plt.title('IMG HSV')
plt.subplot(453), plt.imshow(exe4_img_hls), plt.title('IMG HLS')
plt.show()
```



### Exercício 5

Utilize técnicas de processamento de imagens para:

1. Carregar uma imagem colorida.
2. Aumentar o tamanho da imagem (amostragem) em 50%, por exemplo, uma imagem de 100x100 pixels deverá ser alterada (escalonada) para 150x150 pixels.
3. Apresentar a imagem originalmente carregada, bem como, o resultado após a alteração de escala.

```
In [39]: # 1. Carregar uma imagem colorida.
exe5_img1 = cv.imread('frymire.png')
exe5_img1 = cv.cvtColor(exe5_img1, cv.COLOR_BGR2RGB)
```

```
In [40]: # 2. Aumentar o tamanho da imagem (amostragem) em 50%, por exemplo,
# uma imagem de 100x100 pixels deverá ser alterada (escalonada) para 150x150 pixels.
width = exe5_img1.shape[1] * 1.5
height = exe5_img1.shape[0] * 1.5

exe5_img_scale = cv.resize(exe5_img1, (int(width), int(height)), interpolation=cv.INTER_CUBIC)

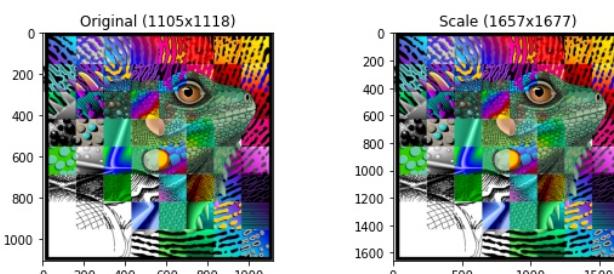
w, h, _ = exe5_img1.shape
exe5_img1_shape = f'{w}x{h}'

w, h, _ = exe5_img_scale.shape
exe5_img_scale_shape = f'{w}x{h}'

f'Original: {exe5_img1_shape}, Scale: {exe5_img_scale_shape}'
```

```
Out[40]: 'Original: (1105x1118), Scale: (1657x1677)'
```

```
In [41]: # 3. Apresentar a imagem originalmente carregada, bem como, o resultado após a alteração de escala.
plt.figure(figsize=(25, 20))
plt.subplot(551), plt.imshow(exe5_img1), plt.title(f'Original {exe5_img1_shape}')
plt.subplot(552), plt.imshow(exe5_img_scale), plt.title(f'Scale {exe5_img_scale_shape}')
plt.show()
```



## Exercício 6

Utilize técnicas de processamento de imagens para:

1. Carregar uma imagem colorida e converter para tons de cinza.
2. Realizar ajuste de brilho para demonstrar qual o resultado do histograma de uma imagem mais brilhante (ou mais clara).
3. Realizar ajuste de brilho para demonstrar qual o resultado do histograma de uma imagem mais escura.
4. Demonstrar o uso da técnica de equalização de histograma.
5. Apresentar os resultados.

```
In [42]: # 1. Carregar uma imagem colorida e converter para tons de cinza.  
exe6_img1 = cv.imread('frymire.png')  
exe6_img1 = cv.cvtColor(exe6_img1, cv.COLOR_BGR2RGB)  
exe6_img_gray = cv.cvtColor(exe6_img1, cv.COLOR_BGR2GRAY)
```

```
In [43]: # 2. Realizar ajuste de brilho para demonstrar qual o resultado  
# do histograma de uma imagem mais brilhante (ou mais clara).  
alpha = 1.5 # Contrast control (1.0 - 3.0)  
beta = 80 # Brightness control (0 - 100)  
  
exe6_img_bright = cv.convertScaleAbs(exe6_img_gray, alpha=alpha, beta=beta)  
  
exe6_hist1_original = cv.calcHist([exe6_img_gray], [0], None, [256], [0, 255])  
exe6_hist2_bright = cv.calcHist([exe6_img_bright], [0], None, [256], [0, 255])
```

```
In [44]: # 3. Realizar ajuste de brilho para demonstrar qual o resultado do histograma de uma imagem mais escura.  
exe6_img_bright_ajust = cv.subtract(exe6_img_gray, 100)  
  
exe6_hist1_bright_ajust = cv.calcHist([exe6_img_bright_ajust], [0], None, [256], [0, 255])
```

```
In [45]: # 4. Demonstrar o uso da técnica de equalização de histograma.  
exe6_img_gray_eq = cv.equalizeHist(exe6_img_gray)  
exe6_img_bright_eq = cv.equalizeHist(exe6_img_bright)  
  
exe6_hist1_gray_eq = cv.calcHist([exe6_img_gray_eq], [0], None, [256], [0, 255])  
exe6_hist2_bright_eq = cv.calcHist([exe6_img_bright_eq], [0], None, [256], [0, 255])
```

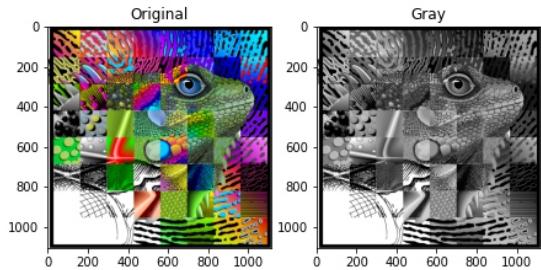
```
In [46]: # 5. Apresentar os resultados.
print('# 1. ---')
plt.figure(figsize=(15, 10))
plt.subplot(141), plt.imshow(exe6_img1), plt.title(f'Original')
plt.subplot(142), plt.imshow(exe6_img_gray, cmap='gray'), plt.title(f'Gray')
plt.show()

print('\n# 2. ---')
plt.figure(figsize=(15, 10))
plt.subplot(541), plt.imshow(exe6_img_gray, cmap='gray'), plt.title(f'Original')
plt.subplot(542), plt.plot(exe6_hist1_original, '-', color='blue'), plt.xlim(0, 255), plt.ylim(0, max(exe6_hist1_original) + 100)
plt.subplot(543), plt.imshow(exe6_img_bright, cmap='gray'), plt.title(f'Brightness')
plt.subplot(544), plt.plot(exe6_hist2_bright, '-', color='red'), plt.xlim(0, 255), plt.ylim(0, max(exe6_hist2_bright) + 100)
plt.show()

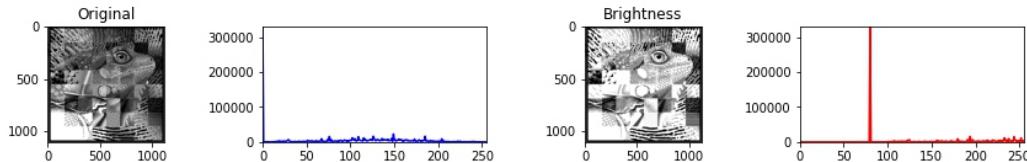
print('\n# 3. ---')
plt.figure(figsize=(15, 10))
plt.subplot(541), plt.imshow(exe6_img_bright, cmap='gray'), plt.title(f'Brightness')
plt.subplot(542), plt.plot(exe6_hist2_bright, '-', color='red'), plt.xlim(0, 255), plt.ylim(0, max(exe6_hist2_bright) + 100)
plt.subplot(543), plt.imshow(exe6_img_bright_ajust, cmap='gray'), plt.title(f'Original')
plt.subplot(544), plt.plot(exe6_hist1_bright_ajust, '-', color='blue'), plt.xlim(0, 255), plt.ylim(0, max(exe6_hist1_bright_ajust) + 100)
plt.show()

print('\n# 4. ---')
plt.figure(figsize=(15, 10))
plt.subplot(541), plt.imshow(exe6_img_gray_eq, cmap='gray'), plt.title(f'Original')
plt.subplot(542), plt.plot(exe6_hist1_gray_eq, '-', color='blue'), plt.xlim(0, 255), plt.ylim(0, max(exe6_hist1_gray_eq) + 100)
plt.subplot(543), plt.imshow(exe6_img_bright_eq, cmap='gray'), plt.title(f'Brightness')
plt.subplot(544), plt.plot(exe6_hist2_bright_eq, '-', color='red'), plt.xlim(0, 255), plt.ylim(0, max(exe6_hist2_bright_eq) + 100)
plt.show()
```

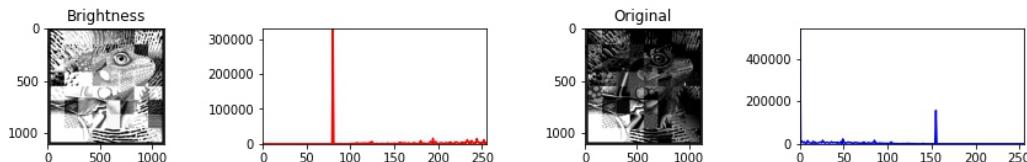
# 1. ---



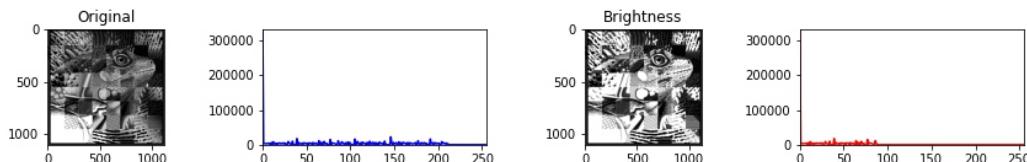
# 2. ---



# 3. ---



# 4. ---



## LINK COLAB:

<https://colab.research.google.com/drive/12O10mQcU56Yit5yFoFk2JMjJqEx9UUZ?usp=sharing>