

FIAP

1. Histograma de Imagens Digitais

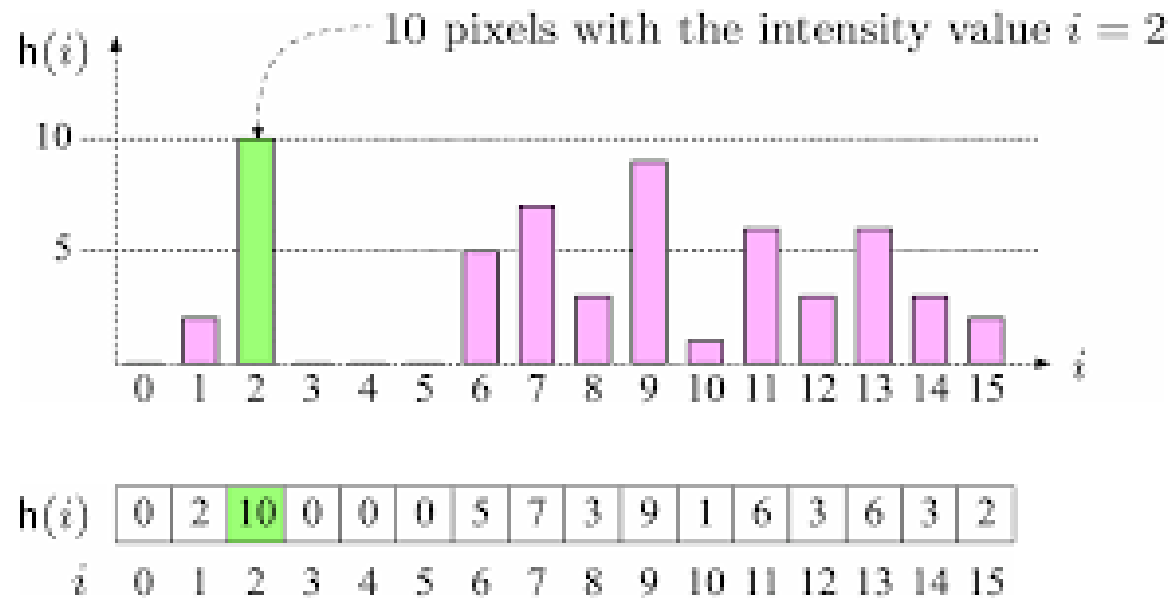
Objetivos da aula:

- Histograma e equalização de histograma
- Segmentação com auxílio do histograma

Histograma

- Distribuição discreta de probabilidade que fornece informações para realce e análise de imagens.
- Um histograma pode nos ajudar, ele plota em um gráfico de frequência as componentes de cores (r,g,b ou gray) da imagem.

Exemplo 1



Exemplo 2

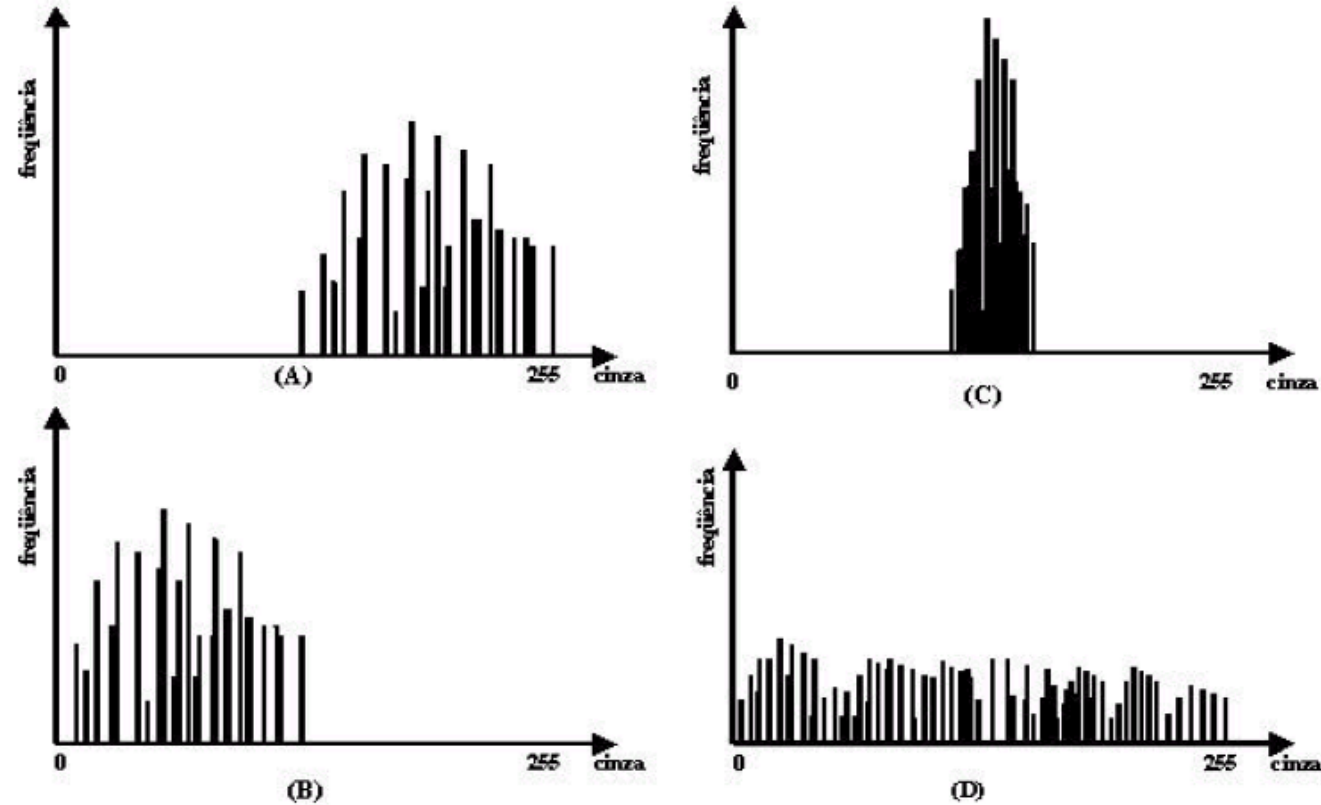
cinza.

6	4	4	6	6
5	0	0	0	5
3	1	1	1	3
2	2	2	2	1
7	7	7	7	7

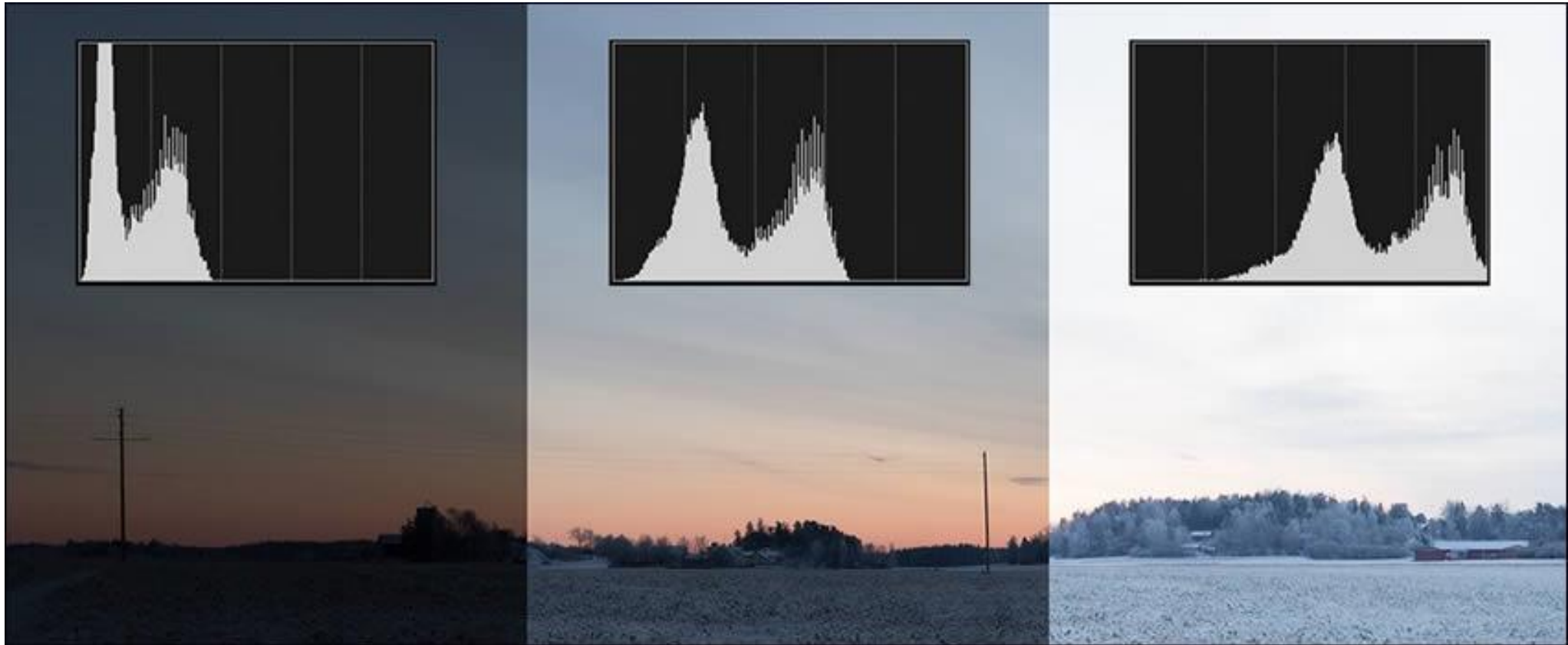
Histograma H =>

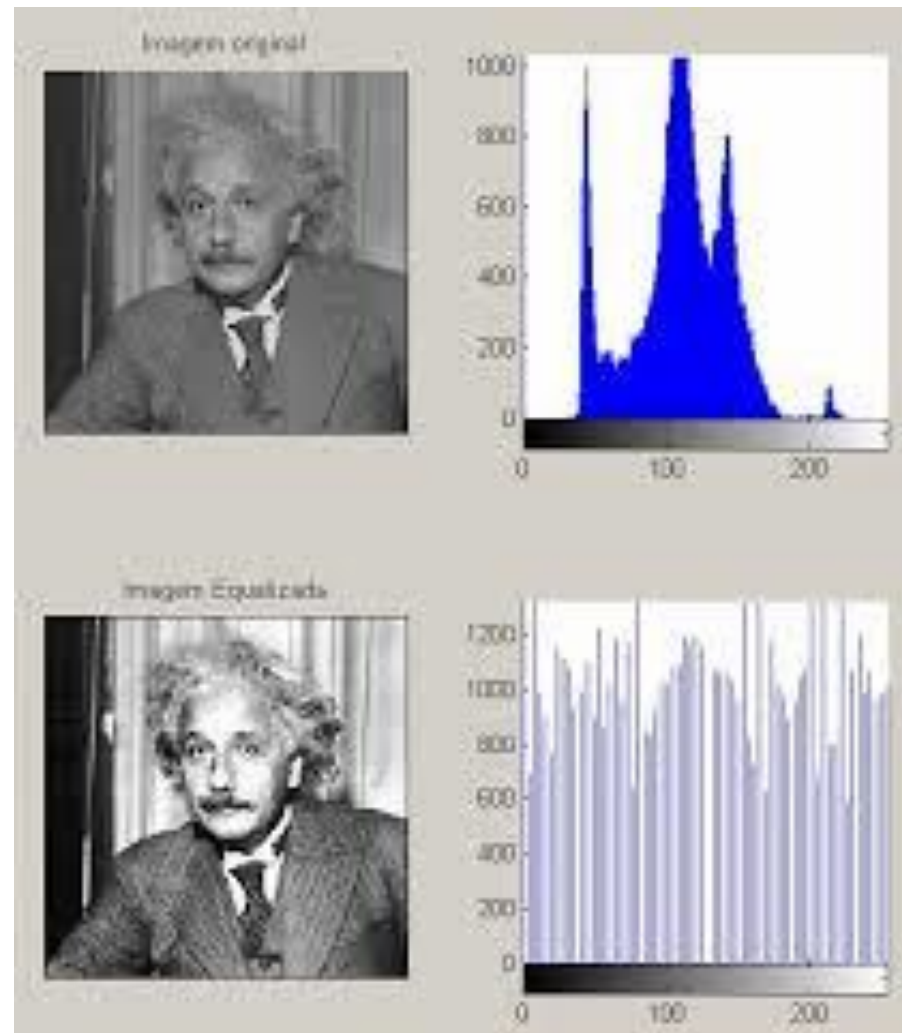
5								
4								
3								
2								
1								
X	0	1	2	3	4	5	6	7

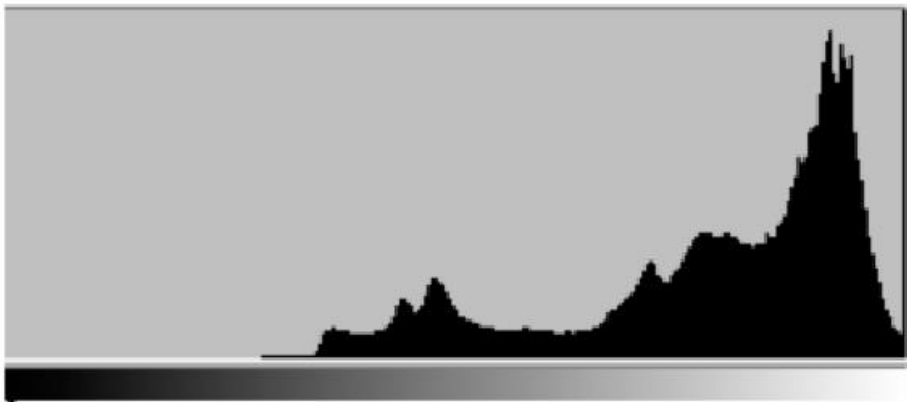
Nível de cinza	0	1	2	3	4	5	6	7
Quantidade	3	4	4	2	2	2	3	5



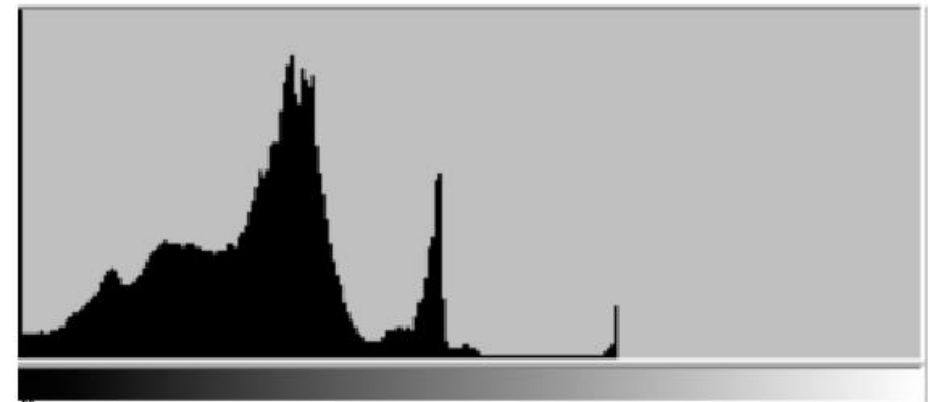
- (A) Imagem clara.
 (B) Imagem escura.
 (C) Imagem com baixo contraste.
 (D) Imagem com alto contraste.



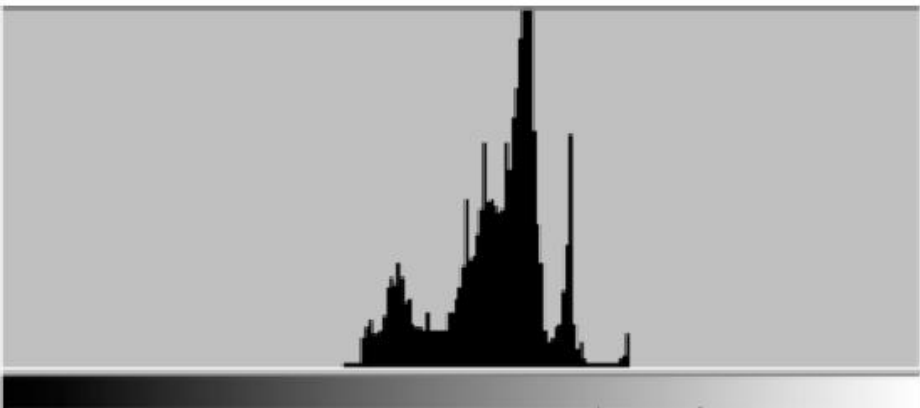




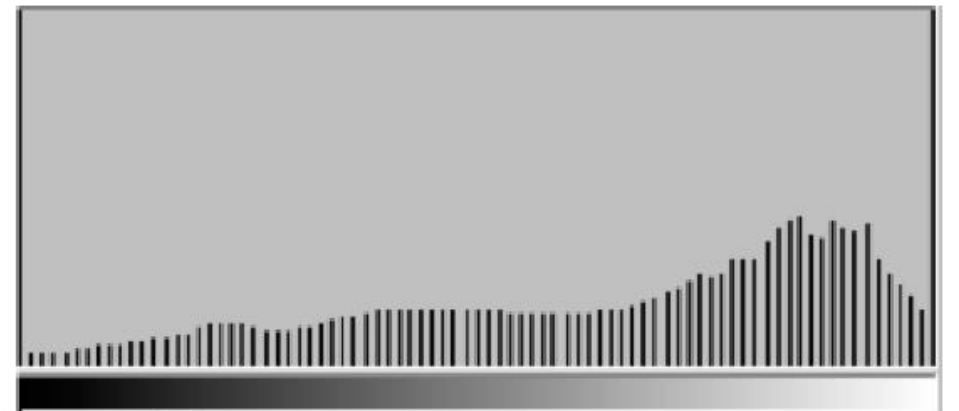
• Alta luminosidade



• Baixa luminosidade



- pouco contraste



- bom contraste

```
%matplotlib inline
import cv2 as cv
from matplotlib import pyplot as plt
import numpy as np

print ("OpenCV Version : %s " % cv2.__version__)
```

```
img = cv2.imread("lenna.jpg")  
img = cv.cvtColor(img, cv.COLOR_BGR2RGB)  
plt.imshow(img); plt.show
```

```
plt.hist(img.ravel(), 256, [0, 256]); plt.show()
```

```
img = cv2.imread("fusca.png", cv2.IMREAD_GRAYSCALE)
plt.imshow(img, cmap="Greys_r", vmin=0, vmax=255)
```

```
plt.hist(img.ravel(), 256, [0, 256]); plt.show()
```



```
img_eq = cv2.equalizeHist(img)
plt.imshow(3*img_eq, cmap="Greys_r", vmin=0, vmax=255)
```

```
plt.hist(3*img_eq.ravel(), 256, [0, 256]); plt.show()
```

IMAGEM COLORIDA

```
imagem = cv2.imread("bola.png")  
image = cv2.cvtColor(imagem, cv2.COLOR_BGR2RGB)  
plt.imshow(image, vmin=0, vmax=255); plt.show()  
plt.hist(image.ravel(), 256, [0, 256]); plt.show()
```

#histograma Vermelho

```
#histograma Vermelho  
plt.imshow(image[:, :, 0], cmap="gray", vmin=0, vmax=255); plt.show()  
plt.hist(image[:, :, 0].ravel(), 256, [0, 256]); plt.show()
```

Histograma Verde

```
# Histograma Verde
plt.imshow(image[:, :, 1], cmap="Greys_r", vmin=0, vmax=255); plt.show()
plt.hist(image[:, :, 1].ravel(), 256, [0, 256]); plt.show()
```

Histograma Azul

```
# Histograma Azul
plt.imshow(image[:, :, 2], cmap="Greys_r", vmin=0, vmax=255); plt.show()
plt.hist(image[:, :, 2].ravel(), 256, [0, 256]); plt.show()
```

```
image2 = image.copy()
gray_r = image2[:, :, 0]
gray_g = image2[:, :, 1]
gray_b = image2[:, :, 2]

img_bola = image2.copy()

for y in range(0, image2.shape[0]):
    for x in range(0, image2.shape[1]):

        if gray_g[y][x] <= 230:
            img_bola[y][x] = 0
        if gray_b[y][x] >= 240:
            img_bola[y][x] = 0

plt.imshow(img_bola, interpolation="none")
plt.show()
```

Desafio 12

Faça a segmentação da bolinha de cor laranja. Dica use 2 canais de cores para conseguir segmentar.

Desafio 13

Faça a segmentação da bolinha para a imagem "bolinha.png".

```
imagem = cv2.imread("bolinha.png")  
image = cv2.cvtColor(imagem, cv2.COLOR_BGR2RGB)  
plt.imshow(image, vmin=0, vmax=255); plt.show()
```