UNIVERSIDADE PRESBITERIANA MACKENZIE
FACULDADE DE COMPUTAÇÃO E INFORMÁTICA
COMPUTAÇÃO VISUAL - 07G

Aluno: Daniel Faia Monteiro da Silva
TIA: 31932029

1. **Como é possível realizar a limiarização de uma imagem usando Python e scikit-image.**

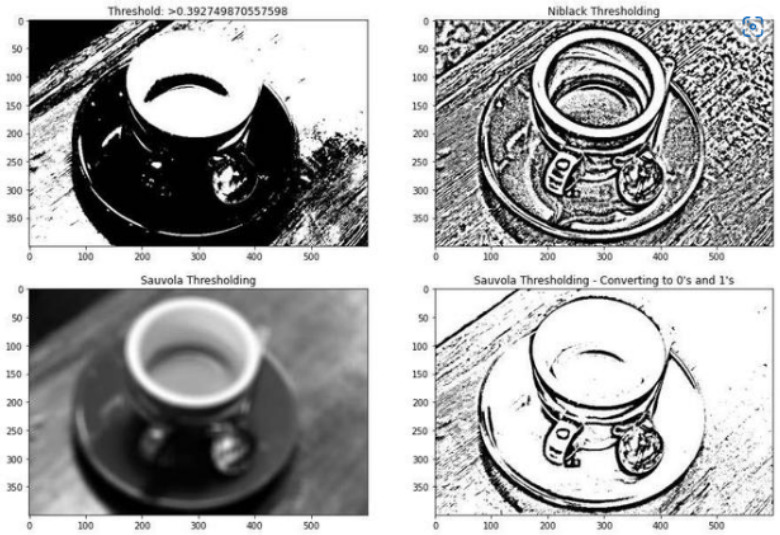
   Existem 3 técnicas de limiarização utilizando Python e scikit-image: a técnica de Niblack, Sauvola e Otsu. As duas primeiras alteram o limite dependendo da média local e do desvio padrão para cada pixel em uma janela deslizante, enquanto a terceira funciona iterando todos os valores de limiar possíveis e calculando uma medida de dispersão para os pontos de amostra em ambos os lados do limiar.

   Para realizar a técnica de limiarização de Otsu, é necessário implementar a função **skimage.filters.threshold_otsu()**, que retorna o valor limite com base no método de Otsu.

   A função **skimage.filters.threshold_niblack()** é uma função de limite local que retorna um valor de limite para cada pixel com base no método de Niblack.

   A função **skimage.filters.threshold_sauvola()** é uma função de limite local que retorna um valor de limite para cada pixel com base no método de Sauvola.

   Tanto a função de Niblack como a de Sauvola retornam uma máscara de limiar igual à forma da imagem.



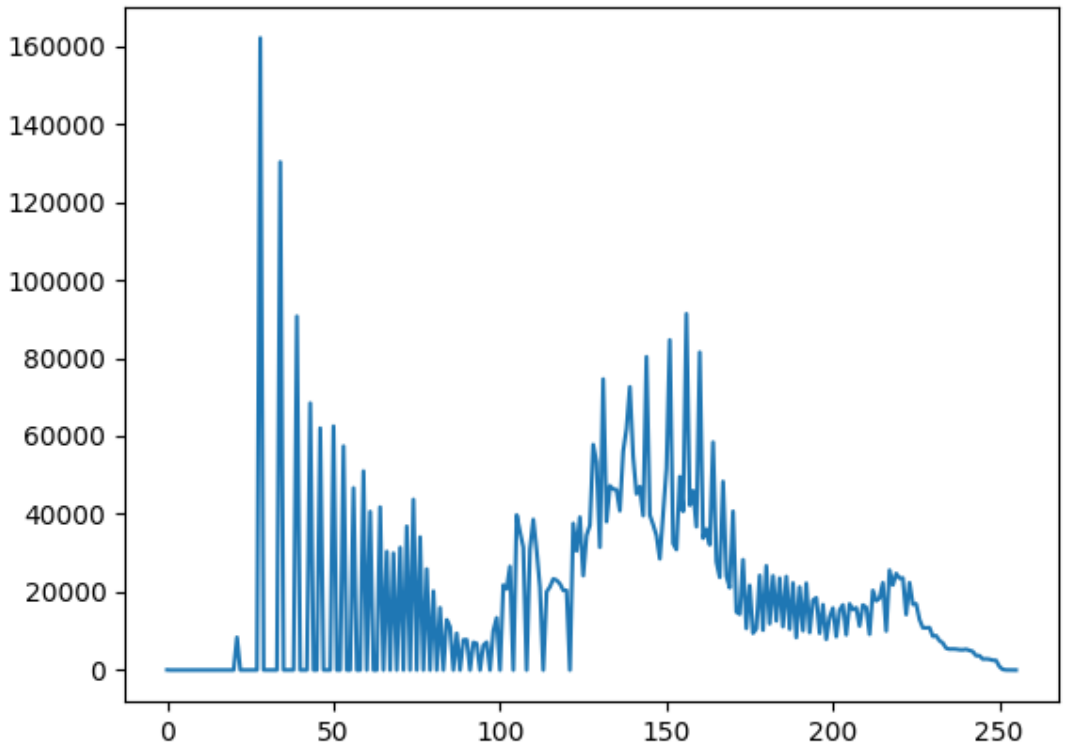
2. **Como é possível plotar o histograma de uma imagem tons de cinza usando Python, scikit-image e matplotlib.**

Para encontrar o histograma de uma imagem, é necessário utilizar a função **hist()** de numpy. Com ela, é possível descobrir se há mais partes claras ou mais partes escuras naquela imagem.

Em seguida, é feito o cálculo do histograma utilizando a função **calcHist(),** que recebe como parâmetro a imagem, canais, máscara, intervalos e tamanho do histograma.

Por fim, é feito a plotagem do histograma assim como mostrado no exemplo abaixo:

```
import cv2
from matplotlib import pyplot as plt
img = cv2.imread('ex.jpg',0)
histr = cv2.calcHist([img],[0],None,[256],[0,256])
plt.plot(histr)
plt.show()
```



3. **Como é possível plotar o histograma de uma imagem colorida (um histograma por canal de cor) usando Python, scikit-image e matplotlib.**

Assim como é possível encontrar o histograma de uma imagem em tons de cinza, também é possível fazer o mesmo com uma imagem colorida.
Calcula-se o histograma com a função cv2.calcHist e normaliza-se as imagens colocando todo o vetor em uma dimensão (numpy.ndarray.flatten).

Segue abaixo um exemplo de como encontrar o histograma de uma imagem colorida:

```
#extrair um histograma de cores RGB da imagem,
# usando 8 caixas por canal, normalizar e atualizar o índice
hist = cv2.calcHist([image], [0, 1], None, [8, 8],
                    [0, 256, 0, 256])
hist = cv2.normalize(hist, hist).flatten()
index[filename] = hist
```

Para plotar a imagem, pode-se utilizar o mesmo método utilizado na imagem em tons de cinza:

```
import cv2
from matplotlib import pyplot as plt
img = cv2.imread('ex.jpg',0)
histr = cv2.calcHist([img],[0],None,[256],[0,256])
plt.plot(histr)
plt.show()
```

4. **Como é possível equalizar o histograma de uma imagem usando Python e scikit-image.**

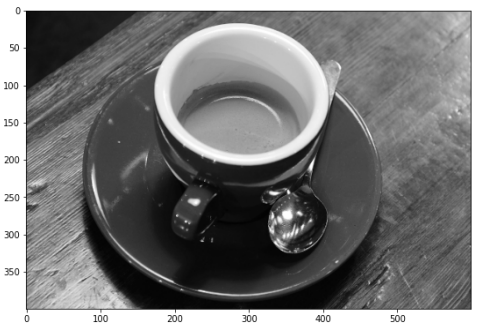
O primeiro passo é realizar um cálculo probabilístico em cima do histograma, ou seja, fazer a divisão de quantas vezes o valor apareceu pelo número total de pixels na imagem.

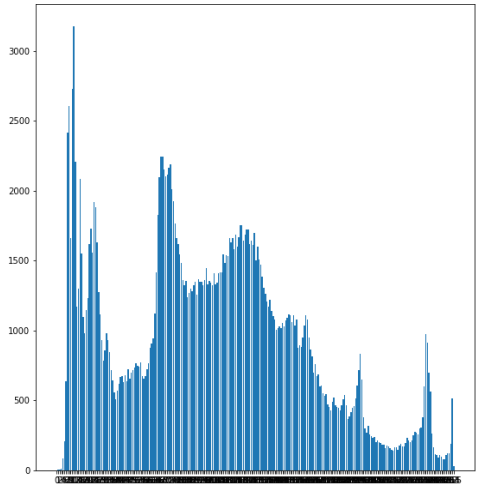
Em seguida, deve-se calcular a probabilidade acumulada, em que para cada iteração o valor do histograma é somado à probabilidade acumulada das iterações anteriores.

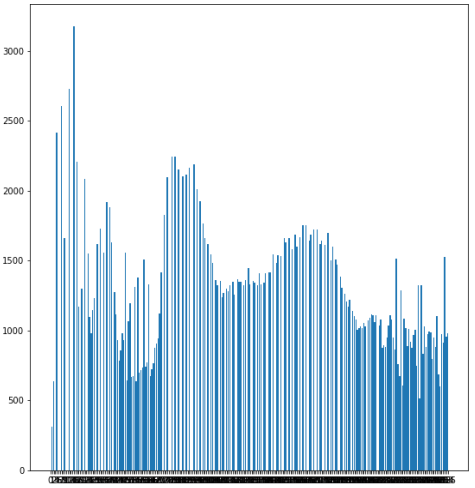
Com as duas probabilidades em mãos, é possível fazer o cálculo dos novos valores de cinza da imagem, ou seja, dado um pixel na posição (x,y) com nível de intensidade z, qual será seu novo valor de intensidade para que o histograma resultante seja equalizado.

No próximo passo, calculamos um novo objeto que irá mapear os respectivos valores de cinza em novos valores equalizados e, por fim, aplicamos os novos valores na imagem original.

As ilustrações abaixo mostram a diferença entre as imagens e os histogramas antes e depois de serem equalizados:

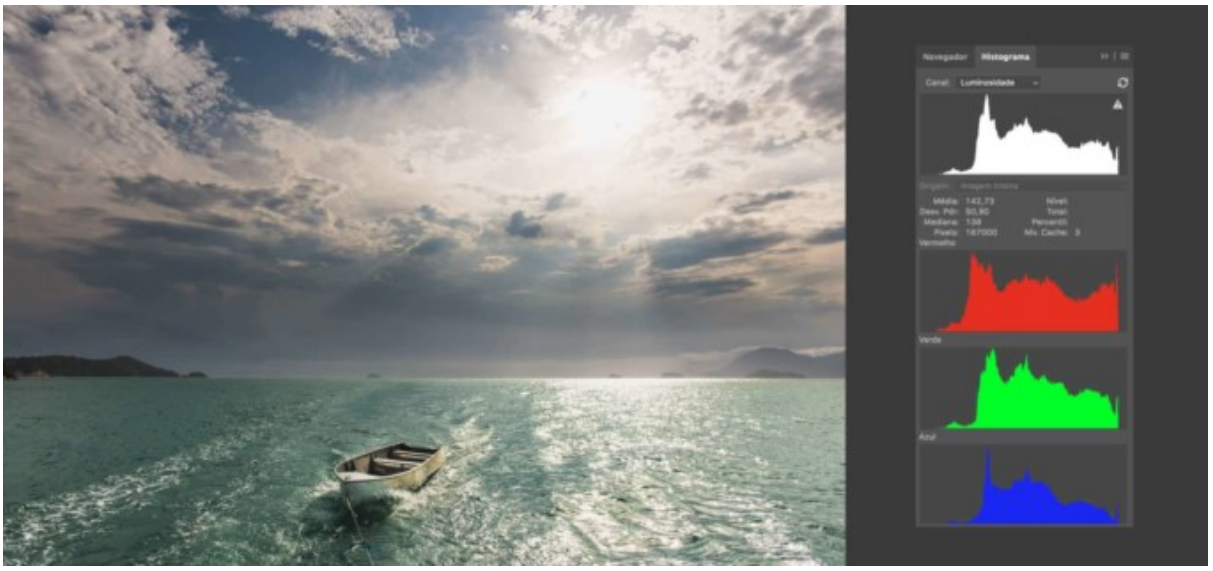




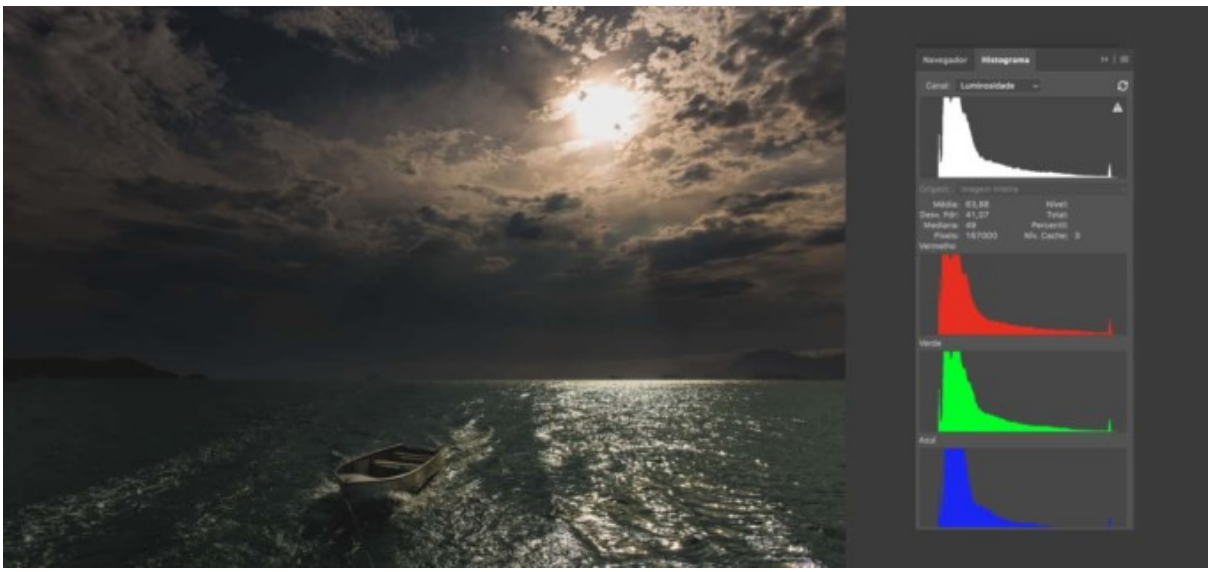


5. **Como é possível detectar (concluir) que uma foto está subexposta ou que está superexposta, analisando o histograma.**

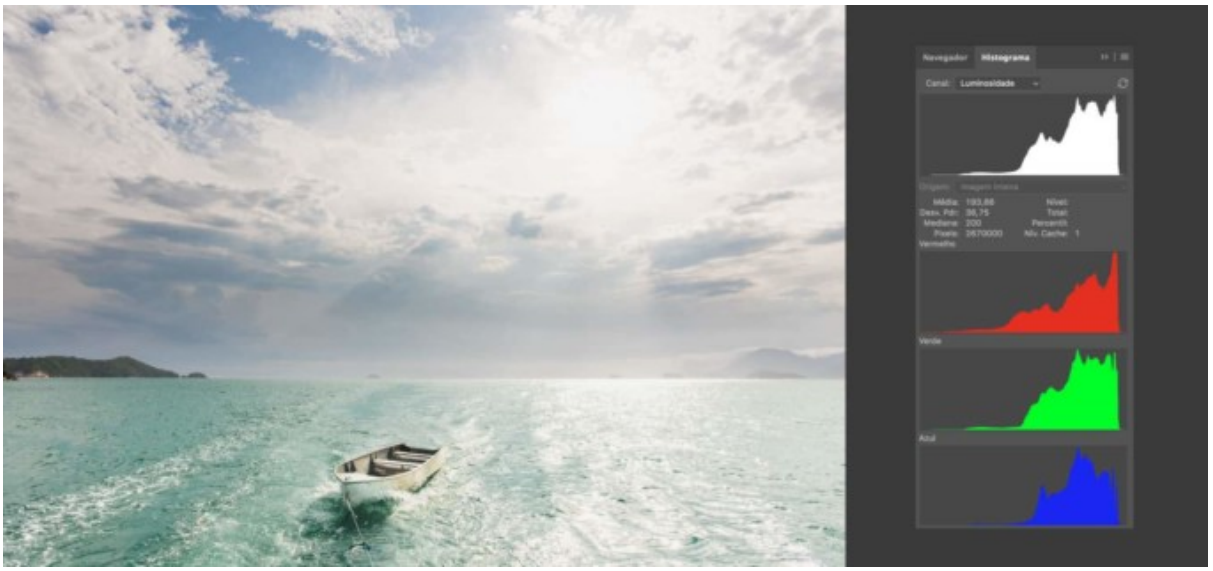
É possível descobrir se uma imagem está subexposta ou superexposta analisando a posição dos dados presentes no histograma. Utilizando a imagem abaixo como exemplo, é possível perceber que o histograma está bem equilibrado em ambos os lados.



Na imagem abaixo, é possível perceber que os dados do histograma estão **concentrados na parte esquerda** dele, indicando que a imagem está **subexposta**, ou seja, bem mais escura que o normal.



Neste outro exemplo, percebe-se que os dados do histograma estão **concentrados na parte direita**, ou seja, a imagem está muito mais clara que o normal, indicando que está **superexposta.**



6. **Como é possível detectar (concluir) se uma imagem está com baixo contraste ou alto contraste, analisando o histograma.**

Histogramas largos são típicos de cenas com bastante contraste, enquanto histogramas estreitos são de imagens com menos contraste e que podem aparentar achatadas ou sem graça.
Além disso, fotos tiradas em condição de neblina ou fumaça terão baixo contraste; fotos tiradas sob sol forte, por outro lado, terão contraste muito mais alto.

O contraste pode ter um impacto visual muito grande ao enfatizar texturas, como mostrado na imagem acima. O alto contraste da água tem sombras mais profundas e brilhos mais pronunciados, criando texturas que saltam aos olhos de quem as observa.

REFERÊNCIAS

Acervo Lima. **SEGMENTAÇÃO DE IMAGENS USANDO O MÓDULO SCIKIT-IMAGE DO PYTHON.** Disponível em: <Segmentação de imagens usando o módulo scikit-image do Python – Acervo Lima>. Acesso em: 20 Set. 2022.

Acervo Lima. **PROGRAMA OPENCV PYTHON PARA ANALISAR UMA IMAGEM USANDO HISTOGRAMA.** Disponível em: <Programa OpenCV Python para analisar uma imagem usando histograma – Acervo Lima>. Acesso em: 20 Set. 2022.

MEGANHA, Felipe. **Comparando Histogramas das imagens com Python e OpenCV.** Maio, 2021. Disponível em: <Comparando Histogramas das imagens com Python e OpenCV | by Felipe Meganha | Medium>. Acesso em: 20 Set. 2022.

CAVALCANTE, Alvaro. **Equalização de histograma em Python.** Data Hackers. Outubro 2020. Disponível em: <Equalização de histograma em Python | by Alvaro Leandro Cavalcante Carneiro | Data Hackers | Medium>. Acesso em: 20 Set. 2022.

LIMA, William. **Desvendando o Histograma na Fotografia.** Criadores Lab. Disponível em: <Histograma na Fotografia: Aprenda como ler e interpretar(criadoreslab.com.br)>. Acesso em: 20 Set. 2022.

Cambridge in Colour. **HISTOGRAMAS - PARTE 1.** Disponível em: <Histogramas de Imagens–Tons e Contraste(cambridge in colour.com)>. Acesso em: 20 Set. 2022.