

# Manipulação de Imagens no Octave/Matlab

Anny Caroline Correa Chagas  
Ciência da Computação, UERJ

21 de Julho de 2019

## Exercício 1

1. Leia a ajuda para as funções `imadjust` e `imhist`

```
1 help imadjust;  
2 help imhist;
```

2. Leia as imagens `Einstein_low_contrast.png`, `Einstein_med_contrast.png` e `Einstein_high_contrast.png`

```
1 ELC = imread("imgs/Einstein_low_contrast.png");  
2 EMC = imread("imgs/Einstein_med_contrast.png");  
3 EHC = imread("imgs/Einstein_high_contrast.png");
```

3. Apresente cada imagem e seu respectivo histograma em uma figura diferente. Compare os histogramas.

Primeiro foi necessário passar as imagens de RGB para escala em cinza.

```
1 ELC = rgb2gray(ELC);  
2 EMC = rgb2gray(EMC);  
3 EHC = rgb2gray(EHC);
```

Depois, foram geradas as imagens:

```
1 #3.a (assumindo que são 3 imagens finais)  
2 function showImgAndHistSP(IMG)  
3     figure;  
4     subplot(2,1,1);  
5     imshow(IMG);  
6     subplot(2,1,2);  
7     imhist(IMG);  
8 endfunction  
9  
10 showImgAndHistSP(ELC); #Figura 1  
11 showImgAndHistSP(EMC); #Figura 2  
12 showImgAndHistSP(EHC); #Figura 3
```

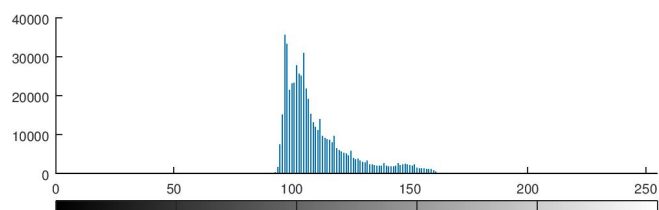
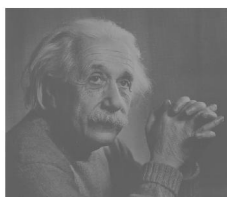


Figura 1: Einstein\_low\_contrast.png e seu histograma

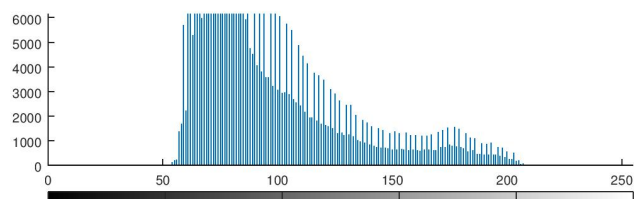
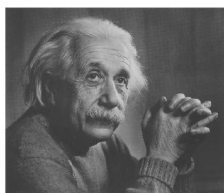


Figura 2: Einstein\_med\_contrast.png e seu histograma

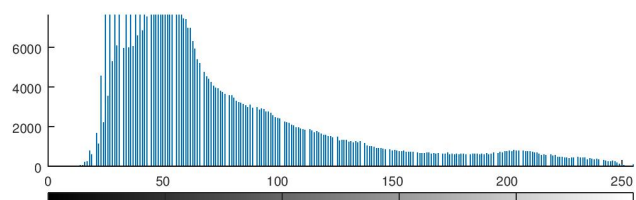
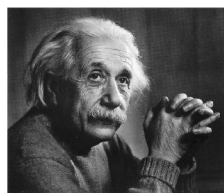


Figura 3: Einstein\_high\_contrast.png e seu histograma

```
1  #3.b (assumindo que são 6 imagens finais)
2  function showImgAndHist(IMG)
3      figure;
4      imshow(IMG);
5
6      figure;
7      imhist(IMG);
8  endfunction
9
10 showImgAndHist(ELC); #Figuras 4 e 5
11 showImgAndHist(EMC); #Figuras 6 e 7
12 showImgAndHist(EHC); #Figuras 8 e 9
```

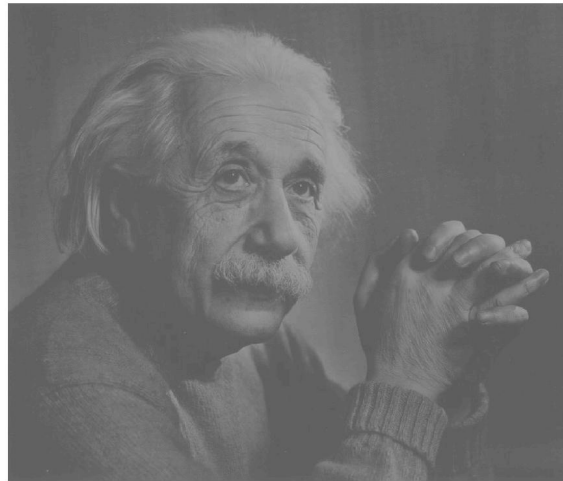


Figura 4: Einstein\_low\_contrast.png

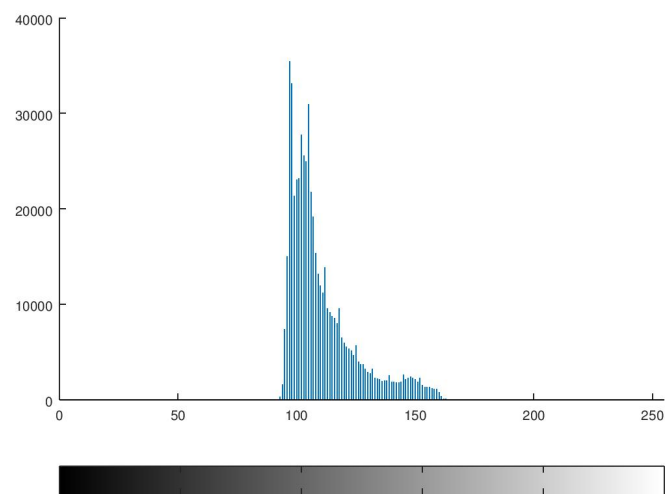


Figura 5: Histograma de Einstein\_low\_contrast.png

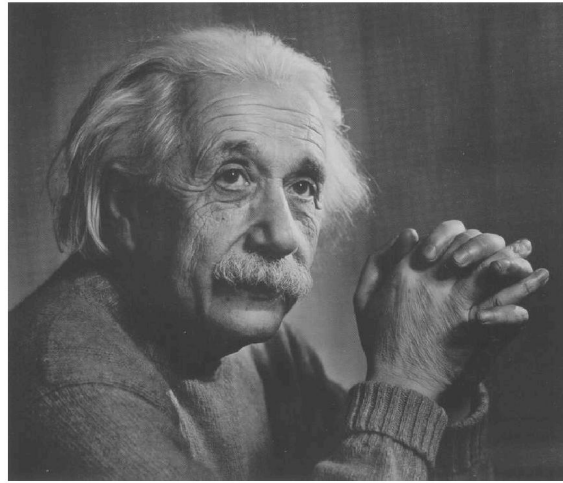


Figura 6: Einstein\_med\_contrast.png

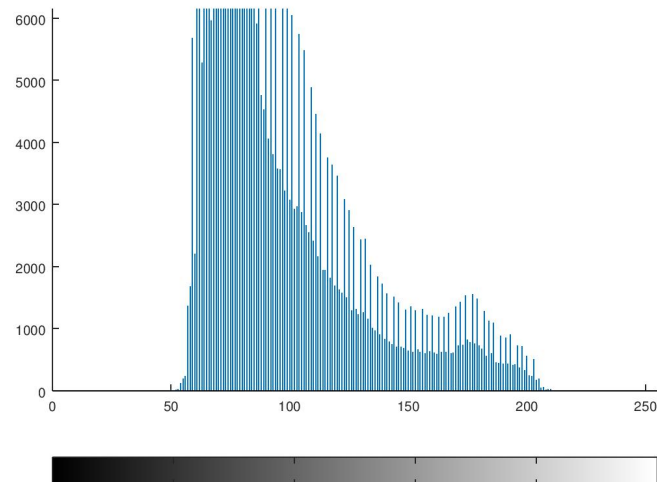


Figura 7: Histograma de Einstein\_med\_contrast.png

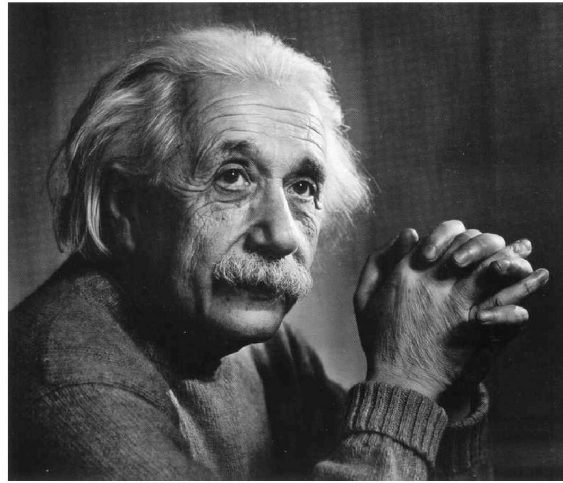


Figura 8: Einstein\_high\_contrast.png

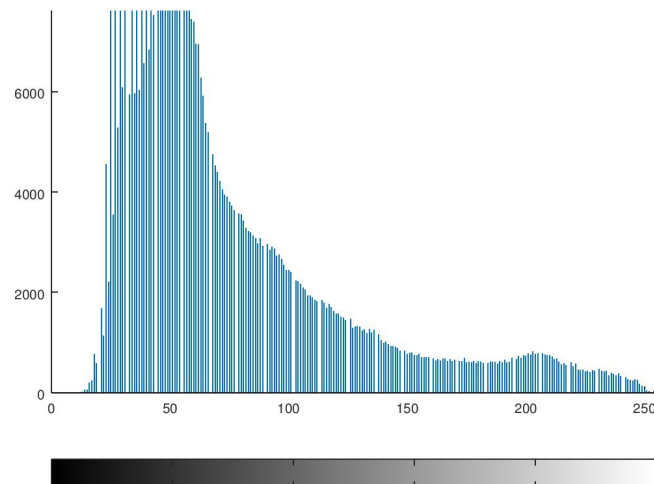


Figura 9: Histograma de Einstein\_high\_contrast.png

4. Utilize a função `imadjust` para melhorar o contraste da imagem `Einstein_low_contrast.png`, de forma a que o histograma da imagem resultante seja parecido com o da imagem `Einstein_high_contrast.png`

```
1 ELC_UINT8 = ELC;  
2 ELC_DOUBLE = im2double(ELC_UINT8);  
3 ELC_AJUST = imadjust(ELC_DOUBLE);
```

5. Apresente a nova imagem e seu histograma em uma única figura (**resultado na Figura 10**).

```
1 showImgAndHistSP(ELC_AJUST); #Figura 10
```

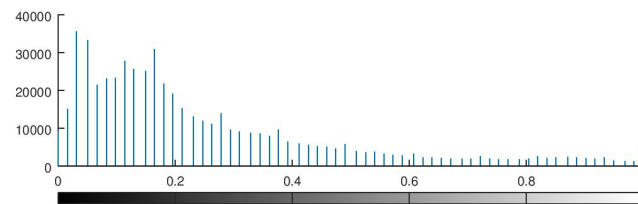
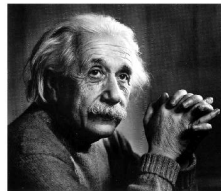


Figura 10: `Einstein_low_contrast.png` com contraste melhorado e seu histograma



## Exercício 2

1. Leia a imagem `leme.bmp`.
2. Utilizando a função `imadjust`, crie uma nova imagem colorida em que as áreas escuras da imagem original fiquem mais claras, mas as áreas claras da imagem original não mudem muito de intensidade na nova imagem.
3. Apresente a imagem original e a nova imagem em uma única figura.

```
1 LEME = imread("imgs/leme.bmp");
2 NOVA = imadjust(im2double(LEME),[0 1], [0 1], 0.5);
3
4 figure
5 subplot(1,2,1);
6 imshow(LEME);
7 subplot(1,2,2);
8 imshow(NOVA);
```



Figura 11: `leme.bmp` com áreas escuras mais claras

## Exercício 3

1. Leia a ajuda para a função `imfilter`.
2. Escrever um script que:
  - (a) Lê uma imagem RGB.
  - (b) Cria uma máscara espacial para suavização linear.
  - (c) Filtra a imagem de entrada (todos os planos) com esta máscara.
  - (d) Mostra a imagem original e filtrada numa única figura.
3. Execute o script para pelo menos duas imagens e apresente os resultados.

```
1 h = fspecial('average', 5);  
2 filteredRGB = imfilter(RGB,h);  
3 figure, subplot(1,2,1), imshow(filteredRGB), subplot(1,2,2), imshow(RGB);
```

Resultado para imagem `arara_full.png`:

```
1 RGB = imread('imgs/arara_full.png');
```



Figura 12: Filtro de suavização para `arara_full.png`

Resultado para imagem `lion.png`:

```
1 RGB = imread('imgs/lion.png');
```



Figura 13: Filtro de suavização para `lion.png`

## Exercício 4

1. Leia a ajuda da função `fspecial` as máscaras disponíveis para filtragem linear.
2. Repita o exercício anterior, usando algumas (ao menos 2) das máscaras disponíveis a partir de `fspecial`.

```
1 RGB = imread('imgs/arara_full.png');
2 #RGB = imread('imgs/lion.png');
3
4 h = fspecial('average', 10);
5 average = imfilter(RGB,h);
6
7 h = fspecial('motion', 20);
8 motion = imfilter(RGB,h);
9
10 h = fspecial('disk', 20);
11 disk = imfilter(RGB,h);
12
13 h = fspecial('gaussian', 20);
14 gaussian = imfilter(RGB,h);
15
16 h = fspecial('log', 20);
17 log = imfilter(RGB,h);
18
19 h = fspecial('laplacian');
20 laplacian = imfilter(RGB,h);
21
22 h = fspecial('unsharp');
23 unsharp = imfilter(RGB,h);
24
25 h = fspecial('motion', 20);
26 motion = imfilter(RGB,h);
27
28 h = fspecial('sobel', 20);
29 sobel = imfilter(RGB,h);
30
31 h = fspecial('prewitt', 20);
32 prewitt = imfilter(RGB,h);
33
34 h = fspecial('kirsch', 20);
35 kirsch = imfilter(RGB,h);
36
37 rows=5;
38 cols=2;
39
40 figure;
41 subplot(rows,cols,1);
42 imshow(average);
43 subplot(rows,cols,2);
44 imshow(motion);
45 subplot(rows,cols,3);
46 imshow(disk);
47 subplot(rows,cols,4);
48 imshow(gaussian);
49 subplot(rows,cols,5);
50 imshow(log);
```

```
51 subplot(rows,cols,6);
52 imshow(laplacian);
53 subplot(rows,cols,7);
54 imshow(unsharp);
55 subplot(rows,cols,8);
56 imshow(sobel);
57 subplot(rows,cols,9);
58 imshow(rewitt);
59 subplot(rows,cols,10);
60 imshow(kirsch);
```

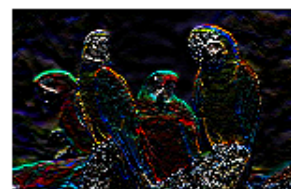
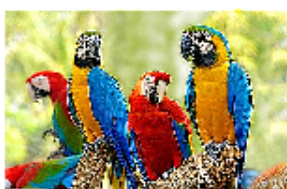


Figura 14: Filtros aplicados à imagem `arara_full.png`



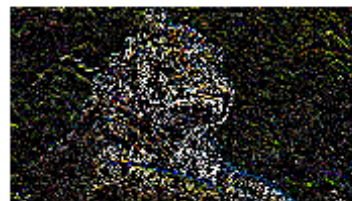
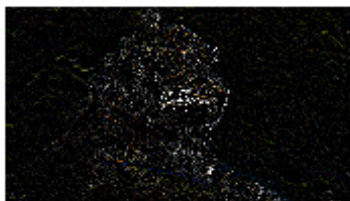
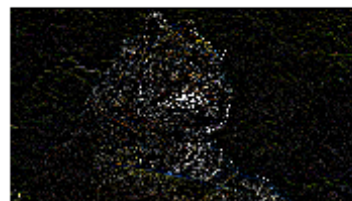
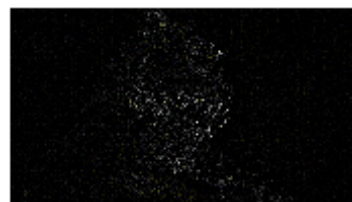


Figura 15: Filtros aplicados à imagem `lion.png`

## Exercício 5

1. Leia a imagem `Coins.png`
2. Com ajuda da função `imhist` identifique um limiar (de intensidade) que melhor separa as moedas do fundo da imagem (background).
3. Com o limiar definido, crie uma imagem binária em que os pixels do background das moedas tenham valores diferentes.
4. Aplique esta máscara binária à imagem original de forma a mudar a cor do background na imagem resultante.
5. Apresente a imagem original e a resultante em uma única figura.

```
1  #1
2  COINS = imread("imgs/Coins.png");
3  COINS = im2double(COINS);
4  showImgAndHist(COINS,'1. Imagem original e seu histograma');
5
6
7  #3 - limiar 0.33 identificado por meio do histograma
8  BIN = COINS > 0.33;
9  showImgAndHist(BIN,'2. Imagem binária e seu histograma');
10
11
12  #4 - mudar a cor de background usando a imagem binária
13  COINS_BIN = BIN .* COINS;
14  showImgAndHist(COINS_BIN, '3. Cor de background alterada usando a imagem binária');
15
16
17  #4 - mudar a cor de background usando diretamente o limiar
18  COINS_LIM = COINS;
19  s = size(COINS_LIM);
20  s = s(1)*s(2);
21  for i = 1:s
22      if COINS_LIM(i) <= 0.33
23          COINS_LIM(i) = 0.2;
24      endif
25  endfor
26  showImgAndHist(COINS_LIM, '4. Cor de background alterada usando diretamente o limiar');
27
28
29  #5
30  figure('name', '5. Apresente a imagem original e a resultante em uma única
31  ↪  figura.','NumberTitle','off');
32  subplot(2,1,1);
33  imshow(COINS);
34  subplot(2,2,3);
35  title('Imagem com cor de background alterada usando imagem binária');
36  imshow(COINS_BIN);
37  subplot(2,2,4);
38  title('Imagem com cor de background alterada usando diretamente o limiar');
39  imshow(COINS_LIM);
```



```

1 #showImgAndHist.m
2
3 function showImgAndHist(IMG, titulo)
4     figure('name', titulo, 'NumberTitle', 'off');
5     subplot(2,1,1);
6     imshow(IMG);
7     subplot(2,1,2);
8     imhist(IMG);
9 endfunction

```

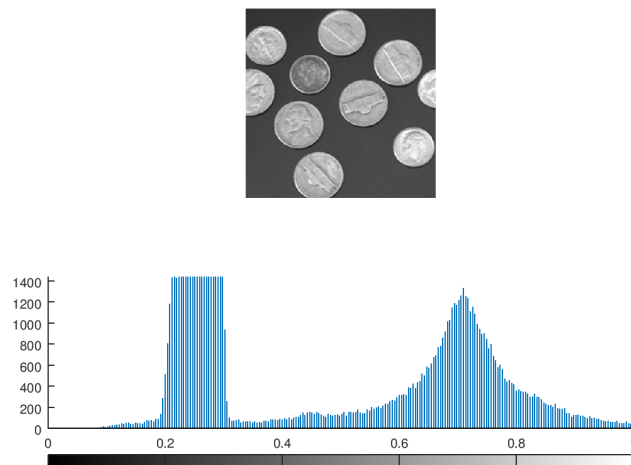


Figura 16: Imagem original e seu histograma

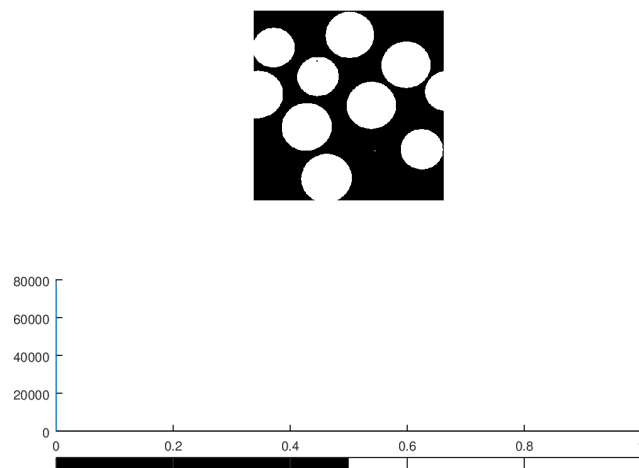


Figura 17: Imagem binária e seu histograma

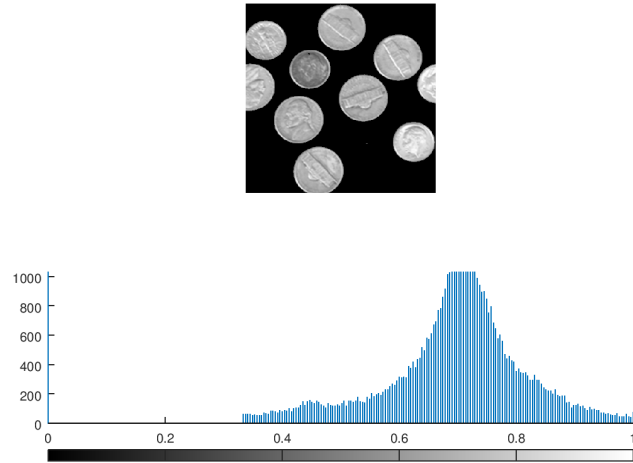


Figura 18: Cor de background alterada usando a imagem binária

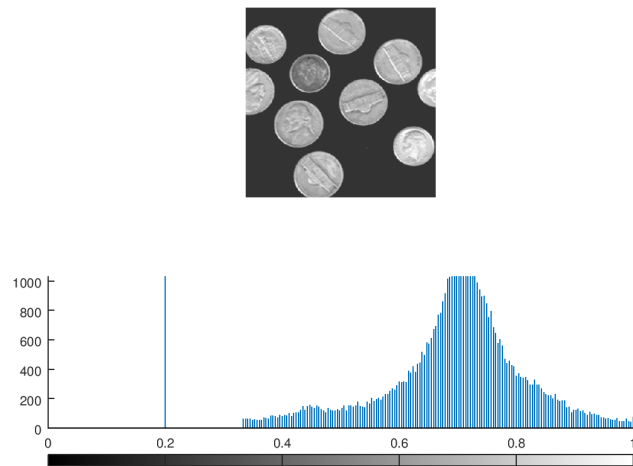


Figura 19: Cor de background alterada usando diretamente o limiar

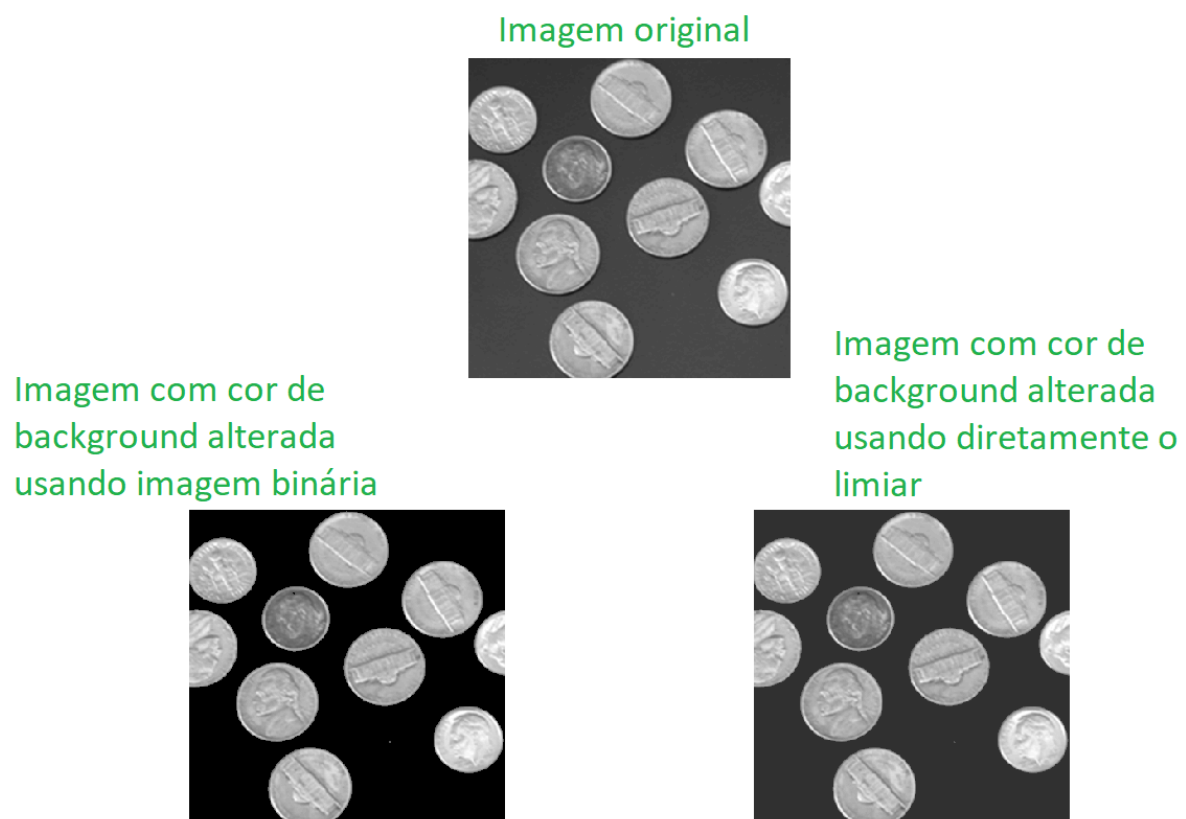


Figura 20: Imagem original e as alteradas