

# IM563

Djalma Santana Malta Neto (203992)

29 de Março de 2023

Para o presente trabalho foi utilizado 4 imagens, sendo 1 capturada por dispositivo móvel, 1 capturada e modificada e 2 retiradas da web, (Imagens: 5, 6, 7 e 8) para mostrar com mais clareza as modificações feitas.

## 1 Questão 01

Redução da resolução da imagem de 50% e 25% através de re-amostragem...

Através de um algoritmo feito na linguagem de programação python (imagem: 2) foi possível obter bons resultados no redimensionamento da imagem, onde foi realizado uma leitura dos pixels e seus "vizinhos" para adquirir a média desses valores e adiciona-lo em outra variável com 50% e 25% do tamanho da original.

Foi realizado, inicialmente, uma transformação da imagem em escala de cinza (imagem 3) onde, através de um loop para percorrer linhas e colunas, foi pego o valor de cada pixel em RGB, aplicado uma redução de 70% desses valores e na respectiva posição feito a média dessas novas medidas e substituído o valor original pelo resultado, dessa forma obteve-se uma imagem com valores iguais na faixa RGB.

Posteriormente essa imagem em escala de cinza foi submetida a outro loop onde foi pego, de cada pixel, os valores de seus "vizinhos" resultando em 9 valores para cada exceto os que se encontram nas bordas, ao retirar a média, esse valor resultante foi adicionado em outra variável cuja dimensão é 50% menor que o da imagem original resultando em uma nova imagem de proporção 174X255 (imagem 4), para adquirir uma imagem com 25% do tamanho foi realizado o mesmo processo na imagem com 50% do tamanho resultando em uma nova imagem de proporção 87X127. A imagem embora semelhante à original, por conta do tamanho reduzido acaba por ficar distorcida ao tentar dar um zoom.

## 2 Questão 02

Separação das camadas RGB e reconstrução de imagens equivalentes em tons de cinza...

Através de um algoritmo feito na linguagem de programação python (imagem: 2) foi realizado a re-amostragem da imagem original seguindo 3 combinações de porcentagem para as cores Vermelho, Verde e Azul respectivamente.

A) 15 - 55 - 30

B) 30 - 15 - 55

C) 55 - 30 - 15

Assim ao aplicar a combinação "A" observa-se que a imagem ficou com uma coloração esverdeada (imagem 9) devido à substituição dos valores das tonalidades por 15% da Vermelha, 55% da Verde e 30% da Azul, dessa forma por ser a que ficou com o valor maior deu à nova imagem a tonalidade verde.

Ao aplicar na imagem a combinação "B" observa-se que a imagem adquiriu uma coloração azulada (imagem 10) devido à substituição dos valores das tonalidades por 30% da Vermelha, 15% da Verde e 55% da Azul, dessa forma por ser a que ficou com o valor maior deu à nova imagem a tonalidade azul.

Ao aplicar na imagem a combinação "C" observa-se que a imagem adquiriu uma coloração avermelhada (imagem 10) devido à substituição dos valores das tonalidades por 55% da Vermelha, 30% da Verde e 15% da Azul, dessa forma por ser a que ficou com o valor maior deu à nova imagem a tonalidade vermelho.

Sendo Assim observa-se que em todas as imagens havia uma mistura na cor predominante, o vermelho não ficou 100% vermelho, da mesma forma o verde e o azul devido à porcentagem de 30% e 15% aplicada em outros espectros de cor fazendo com que a cor predominante recaísse onde foi substituído o valor pelos 55%.

### 3 Questão 03

Equalização da imagem através do escalonamento e translação do seu histograma...  
OBS.: Alguns dados para essa atividade foram retirado de uma publicação feita por Carneiro (2020) e Martins (2015?)

Para realizar o processo de translação do histograma da imagem foi necessário primeiro identificar o valor máximo e mínimo de intensidade, para que possamos saturar a porcentagem e não ultrapassar o valor máximo de 255.

Dessa forma foi realizado um loop para percorrer linhas e colunas da imagem pegando cada valor de intensidade e decrementando o valor mínimo, dessa forma a nossa imagem inicia na intensidade 0, posteriormente é incrementando uma porcentagem na intensidade de cada pixel como mostra a imagem abaixo:

Figure 1: Calculo para equalizar imagem

```
valueMin = min(values)
valueMax = max(values)-valueMin
value = ((255-valueMax)*100)/valueMax/100

# translata a intencidade da imagem de forma que comece em 0
for l in range(0,L):
    for c in range(0,C):
        val = img[l,c]-valueMin
        val = math.ceil(val+(val*value))
        newImage[l,c] = 255 if(val>255) else 0 if(val<0) else val
```

Fonte: Elaborado pelo autor (2023)

Como resultado podemos ver na imagem 12 que obteve um aumento significativo da nitidez dos detalhes e, na imagem 13, temos a comparação dos histogramas da mesma imagem antes e depois do processo.

## 4 Questão 04

Equalização da imagem através da linearização do histograma acumulado... OBS.: Alguns dados para essa atividade foram retirado de uma publicação feita por Carneiro (2020) e Martins (2015?)

Para realizar o processo de equalização através do histograma acumulado primeiramente foi necessário calcular a probabilidade de cada intensidade em uma imagem onde basicamente foi realizado a contagem de quantas vezes aquela intensidade apareceu em toda a imagem, esse valor foi dividido pela quantidade total de pixels na imagem, os valores resultantes foram percorridos em um loop onde a cada interação o valor atual foi substituído pela soma dos valores anteriores dando assim origem a uma lista de valores cumulativos.

Com esses valores foi possível realizar a substituição de uma intensidade pelo resultado cumulativo de sua probabilidade multiplicado pelo valor máximo possível de intensidade resultando em uma imagem mais nítida, podemos ver na comparação dos resultados nas imagens: 14, 15 e 16.

## 5 Questão 05

Extrair as bordas da imagem... OBS.: Alguns dados para essa atividade foram retirado do trabalho de Meechan et al. (2023)

Para o processo de identificação de bordas na imagem foi realizado 4 processos distintos. Inicialmente foi realizado a transformação da imagem em escala de cinza onde obteve-se uma única faixa de cor que varia entre 0 e 255. Posteriormente foi realizado a equalização dessa imagem que se fez necessário devido ao nível de precisão que os testes apresentaram, onde, sem ele, as bordas ficaram

difíceis de identificar devido à variedade de tonalidades presente na imagem, dessa forma a diferença entre as tonalidades se tornaram mais evidentes.

Logo depois foi realizado o processo de Blurred, onde cada pixel passa a carregar características de pixels vizinhos, deixando a imagem com aspecto "embaçado". Por fim é realizado o processo de identificação de bordas, onde a imagem é percorrida e cada pixel é comparado com o "anterior" e o pixel "superior", onde a intensidade desse pixel foi substituída por 255 caso seja menor ou maior que a do pixel comparado e substituído por 0 caso seja extremamente semelhante ao valor. Como resultado (Imagem 17) é possível observar que o processo funciona com um certo nível de satisfação, porem essa precisão varia de acordo com a imagem.

## 6 APÊNDICES

Figure 2: Função para fazer o redimensionamento da imagem

```

4 def redimensionar(imagem):
5     L = imagem.shape[0]
6     C = imagem.shape[1]
7     c = C // 2 if C%2 == 0 else C-1
8     l = L // 2 if L%2 == 0 else L-1
9
10    reducao = 50 # valor em %
11
12    print('Altura: [0]', format(L))
13    print('Largura: [1]', format(C))
14
15    newL = int(L*(reducao/100))
16    newC = int(C*(reducao/100))
17
18    newImage = cv.cvtColor(newImage, cv.COLOR_BGR2GRAY)
19    newImage = cv.cvtColor(newImage, cv.COLOR_GRAY2BGR)
20    newImage = cv.cvtColor(newImage, cv.COLOR_BGR2RGB)
21
22    print('Nova Altura: [0]', format(newL))
23    print('Nova Largura: [1]', format(newC))
24
25    countL = 0
26    countC = 0
27    media = 0
28
29    for l in range(0, L):
30        for c in range(0, C):
31            coordenadas = [
32                (l-1, c-1), (l-1, c), (l-1, c+1), (l, c-1), (l, c), (l, c+1), (l+1, c-1), (l+1, c), (l+1, c+1)
33            ]
34            coord = []
35
36            for x in range(len(coordenadas)):
37                coord = coordenadas[x]
38                if (coord[0] >= 0 and coord[0] < newL and coord[1] < newC):
39                    media = media + int(imagem[coord[0], coord[1]][0])
40
41            newImage[countL, countC] = int(media)
42            media = 0
43            countC = countC + 1
44            countL = countL + 1
45            print(countL)
46
47    # print(media)
48    return newImage

```

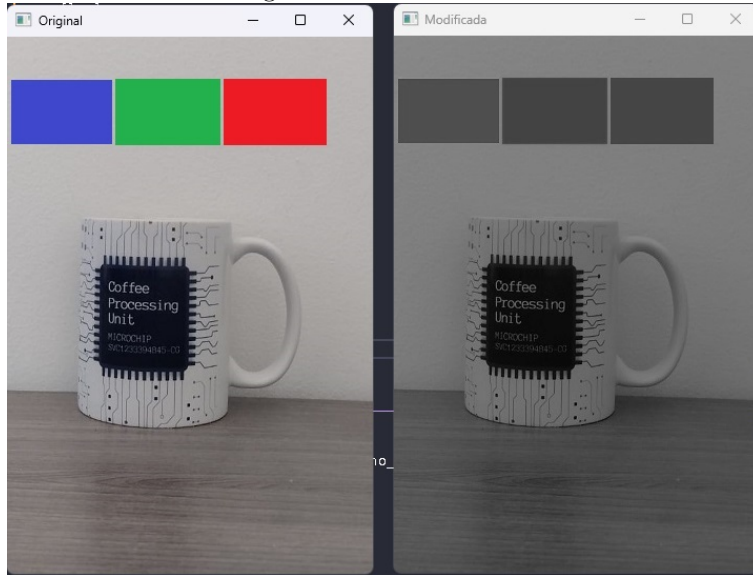
```

51 def escalaCinza(imagem):
52     A = imagem.shape[0]
53     L = imagem.shape[1]
54
55     for l in range(A):
56         for c in range(L):
57             (b, g, r) = imagem[l, c]
58
59             b = b*0.70
60             g = g*0.70
61             r = r*0.70
62
63             gray = (b+g+r)/3
64
65             # imagem[l, c] = (b, g, r)
66             imagem[l, c] = gray
67     return imagem

```

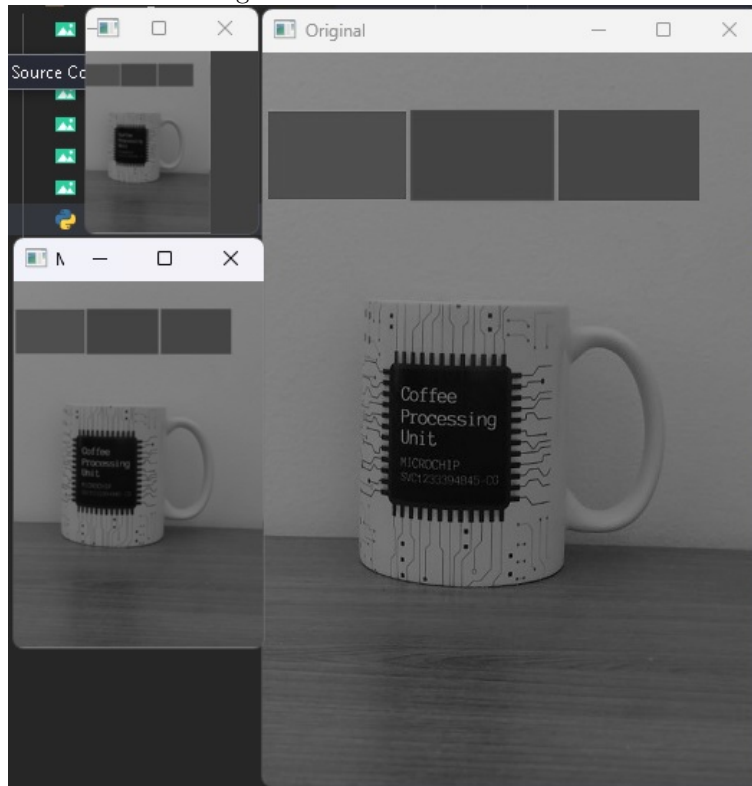
Fonte: Elaborado pelo autor (2023)

Figure 3: Escala de cinza



Fonte: Elaborado pelo autor (2023)

Figure 4: Redimensionamento



Fonte: Elaborado pelo autor (2023)

Figure 5: Imagem com dimensão 510X348



Fonte: Elaborado pelo autor (2023)

Figure 6: Imagem com dimensão 510X348 modificada com quadros nas cores RGB



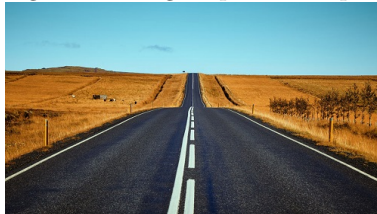
Fonte: Elaborado pelo autor (2023)

Figure 7: Imagem para exemplos



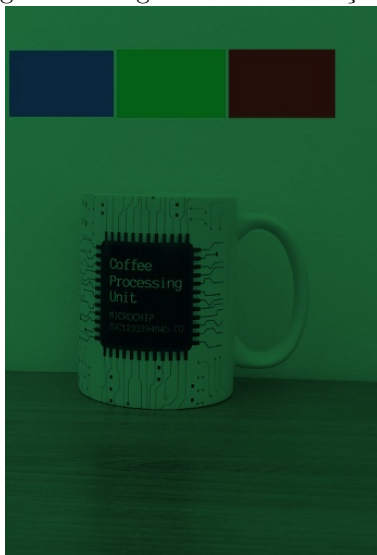
Fonte: [www.significados.com.br](http://www.significados.com.br) (2023)

Figure 8: Imagem para exemplos



Fonte: [www.paroquiadetires.or](http://www.paroquiadetires.or) (2023)

Figure 9: Imagem com combinação A



Fonte: Elaborado pelo autor (2023)



Figure 10: Imagem com combinação B



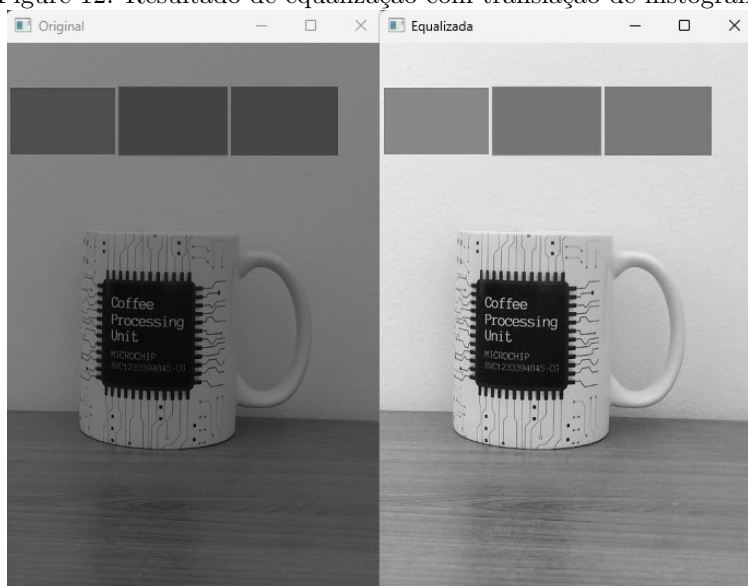
Fonte: Elaborado pelo autor (2023)

Figure 11: Imagem com combinação C



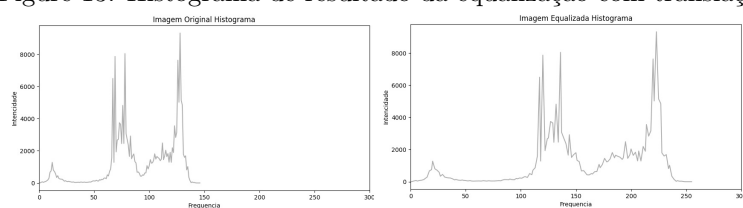
Fonte: Elaborado pelo autor (2023)

Figure 12: Resultado de equalização com translação de histograma



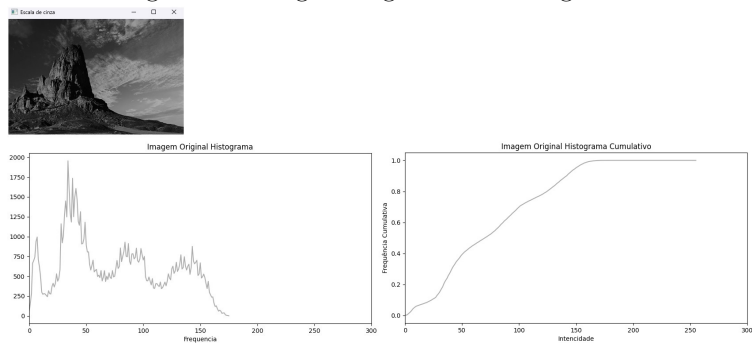
Fonte: Elaborado pelo autor (2023)

Figure 13: Histograma de resultado da equalização com translação



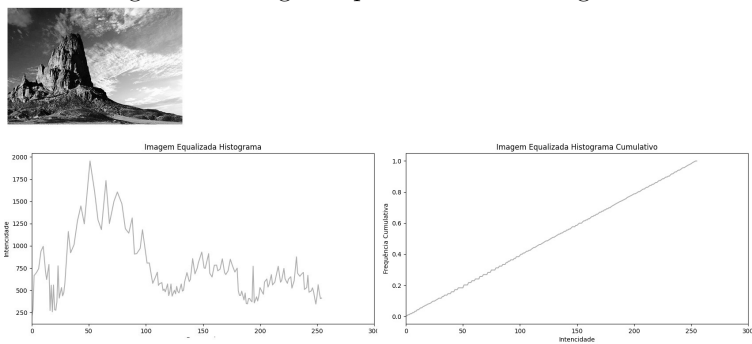
Fonte: Elaborado pelo autor (2023)

Figure 14: Imagem original com histogramas



Fonte: Elaborado pelo autor (2023)

Figure 15: Imagem equalizada com histogramas



Fonte: Elaborado pelo autor (2023)

Figure 16: Comparação entre imagem original e equalizada



Fonte: Elaborado pelo autor (2023)

Figure 17: Resultado do processo de identificação de bordas



Fonte: Elaborado pelo autor (2023)

## References

- CARNEIRO, A. L. C. *Equalização de histograma em Python*. [S.l.], 2020. Disponível em: <https://medium.com/data-hackers/equalizacao-de-histograma-em-python-378830368d60>. Acesso em: 23 de março de 2023.
- MARTINS, A. C. G. *Transformações Geométricas*. [S.l.], 2015? Disponível em: <https://www.sorocaba.unesp.br/Home/Graduacao/EngenhariaAmbiental/antonio/pdi-pos-trasformacoes-geometricas-e-realce-de-imagens-2015.pdf>. Acesso em: 23 de março de 2023.
- MEECHAN, K. et al. *Blurring Images*. [S.l.], 2023. Disponível em: <https://datacarpentry.org/image-processing/06-blurring/>. Acesso em: 29 de março de 2023.