

Universidade do Estado de Santa Catarina UDESC

Transformações sobre pixel

Experimento 1: Estudo sobre contraste

Aluno

Alex Halatiki Vicente

Prof. Responsável

Gilmário

9 de maio de 2023

Introdução

Neste trabalho iremos realizar estudos sobre imagens realizando transformações sobre elas e analisar os resultados obtidos.

Fundamentação

Serão abordados temas como medias, variância e entropia de pixels de imagens alem da implementação do algoritmo para equalização de histogramas.

A média, a variância e a entropia são medidas estatísticas comuns usadas para caracterizar os pixels de uma imagem digital. Cada uma dessas medidas pode fornecer informações valiosas sobre a distribuição dos níveis de intensidade dos pixels em uma imagem.

A média é uma medida de tendência central que representa o valor médio dos níveis de intensidade dos pixels em uma imagem. É calculada somando-se todos os valores de intensidade dos pixels e dividindo o resultado pelo número total de pixels na imagem. A média é útil para caracterizar a iluminação geral de uma imagem e pode ser usada para ajustar o brilho e o contraste da imagem.

A variância é uma medida da dispersão dos níveis de intensidade dos pixels em relação à média. É calculada somando-se a diferença entre cada valor de intensidade de pixel e a média, elevando ao quadrado e dividindo pelo número total de pixels na imagem. Uma variância alta indica que os níveis de intensidade dos pixels estão muito espalhados em torno da média, enquanto uma variância baixa indica que eles estão mais concentrados.

A entropia é uma medida da incerteza ou aleatoriedade dos níveis de intensidade dos pixels em uma imagem. É calculada a partir da distribuição de frequência dos níveis de intensidade dos pixels na imagem. Uma entropia alta indica que a imagem tem uma distribuição uniforme de níveis de intensidade, enquanto uma entropia baixa indica que a distribuição é mais concentrada em torno de um conjunto de valores.

A partir do histograma de uma imagem, que é uma representação gráfica da distribuição de frequência dos níveis de intensidade de uma imagem, é possível aplicar a técnica de processamento de imagem chamada equalização de histograma, que se baseia em uma operação matemática que redistribui os valores de intensidade dos pixels de uma imagem de forma que a distribuição de frequência de cada valor de intensidade se torne mais uniforme, melhorando o contraste de uma imagem.

Etapa Experimental

Questão 1

```
import numpy as np
import matplotlib.pyplot as plt
from skimage import io
from scipy.stats import entropy

def processaImagem(caminho, nome):
    imagem = io.imread(caminho)

    media = np.mean(imagem)
    variancia = np.var(imagem)

    histograma, intervalos = np.histogram(imagem, bins=range(257))
    entropia = entropy(histograma)

    print(f'Imagem {nome}:')
    print("Média: ", media)
    print("Variância: ", variancia)
    print("Entropia: ", entropia)
    print()

    plt.bar(intervalos[:-1], histograma, width=1.0, color='b')
    plt.xlim([0, 255])
    plt.xlabel('Intensidade')
    plt.ylabel('Quantidade')
    plt.savefig(f'./ex1/histogramas/{nome}_histograma.png')

processaImagem('./imagens/figuraClara.jpg', 'figuraClara')
processaImagem('./imagens/figuraEscura.jpg', 'figuraEscura')
processaImagem('./imagens/lena_B.png', 'lena')
```

Com o código acima feito em python, determina-se a media, variância e entropia dos pixels das imagens figuraClara.jpg, figuraEscura.jpg e lena_B.png, além de extrair os histogramas das mesmas.

Questão 2

```
import numpy as np
from skimage import io, color

def equalizaImagem(caminho, nome, formato):
    imagem = io.imread(caminho)

    histograma, intervalos = np.histogram(imagem, bins=8, range=(0, 255))
    histograma = histograma / imagem.size

    roundgk = []

    for i in range(0,8):
        if i != 0:
            histograma[i] += histograma[i-1]
            roundgk.append(round(histograma[i] * 7))

    for i in range(imagem.shape[0]):
        for j in range(imagem.shape[1]):
            indice_intervalo = np.digitize(imagem[i][j], intervalos)

            if(indice_intervalo > 8):
                indice_intervalo = 8

            novo_intervalo = roundgk[indice_intervalo - 1] + 1

            minimo_intervalo = intervalos[indice_intervalo - 1]
            maximo_intervalo = intervalos[indice_intervalo]

            minimo_novo_intervalo = intervalos[novo_intervalo - 1]
            maximo_novo_intervalo = intervalos[novo_intervalo]

            imagem[i][j] = minimo_novo_intervalo + ((imagem[i][j] - minimo_intervalo) / (maximo_intervalo - minimo_intervalo)) * (maximo_novo_intervalo - minimo_novo_intervalo)

    io.imsave(f'./ex2/imagens_equalizadas/{nome}_equalizada.{formato}', imagem)

    print(f'{nome}:')
    print(histograma)
    print(roundgk)
    print(imagem)
    print()

equalizaImagem('./imagens/marilyn.jpg', 'marilyn', 'jpg')
equalizaImagem('./imagens/figuraEscura.jpg', 'figuraEscura', 'jpg')
equalizaImagem('./imagens/figuraClara.jpg', 'figuraClara', 'jpg')
```

Com o código acima feito em python, obtém-se os histogramas das imagens matilyn.jpg, figuraEscura.jpg e figuraClara.jpg e a partir deles, realiza-se a equalização dos mesmos para redistribuir os valores de intensidade dos pixels, com o objetivo de torná-los mais uniformes sobre as imagens.

Questão 3

```
def equalizaImagemRGB(caminho, nome, formato):
    img = cv2.imread(caminho)

    blue = img[:, :, 0]
    green = img[:, :, 1]
    red = img[:, :, 2]

    equalizaMatriz(blue)
    equalizaMatriz(green)
    equalizaMatriz(red)

    imagem_equalizada = cv2.merge([blue, green, red])

    cv2.imwrite(f'./ex3/rgb_equalizadas/{nome}_equalizada.{formato}', imagem_equalizada)

def equalizaImagemYIQ(caminho, nome, formato):
    img = io.imread(caminho)

    img_yiq = color.rgb2yiq(img)

    img_yiq = img_yiq - img_yiq.min()
    img_yiq = img_yiq * 255 / img_yiq.max()
    img_yiq = np.uint8(img_yiq)

    y = img_yiq[:, :, 0]

    equalizaMatriz(y)

    img_yiq[:, :, 0] = y

    img_rgb = color.yiq2rgb(img_yiq)

    img_rgb = img_rgb - img_rgb.min()
    img_rgb = img_rgb * 255 / img_rgb.max()
    img_rgb = np.uint8(img_rgb)

    io.imsave(f'./ex3/yiq_equalizadas/{nome}_equalizada.{formato}', img_rgb)

equalizaImagemRGB('./imagens/outono_LC.png', 'outono_equalizada', 'png')
equalizaImagemRGB('./imagens/predios.jpeg', 'predios_equalizada', 'jpeg')
equalizaImagemYIQ('./imagens/outono_LC.png', 'outono_equalizada', 'png')
equalizaImagemYIQ('./imagens/predios.jpeg', 'predios_equalizada', 'jpeg')]
```

Com o código acima feito em python, submete-se as imagens outono_LC.png e predios.jpeg primeiramente para a função equalizaImagemRGB, que separa os canais red, green e blue, equaliza-os e depois faz a junção novamente para formar a imagem equalizada. Depois submetemos as mesmas imagens para a função equalizaImagemYIQ, que converte as imagens em YIQ, equaliza o canal Y e depois converte novamente para RGB. As equalizações são realizadas pela função exibida na imagem de código abaixo.

```
import cv2
import numpy as np
from skimage import io, color

def equalizaMatriz(matriz):
    histograma, intervalos = np.histogram(matriz, bins=8, range=(0, 255))
    histograma = histograma / matriz.size

    roundgk = []

    for i in range(0,8):
        if i != 0:
            histograma[i] += histograma[i-1]
            roundgk.append(round(histograma[i] * 7))

    for i in range(matriz.shape[0]):
        for j in range(matriz.shape[1]):
            indice_intervalo = np.digitize(matriz[i][j], intervalos)

            if(indice_intervalo > 8):
                indice_intervalo = 8

            novo_intervalo = roundgk[indice_intervalo - 1] + 1

            minimo_intervalo = intervalos[indice_intervalo - 1]
            maximo_intervalo = intervalos[indice_intervalo]

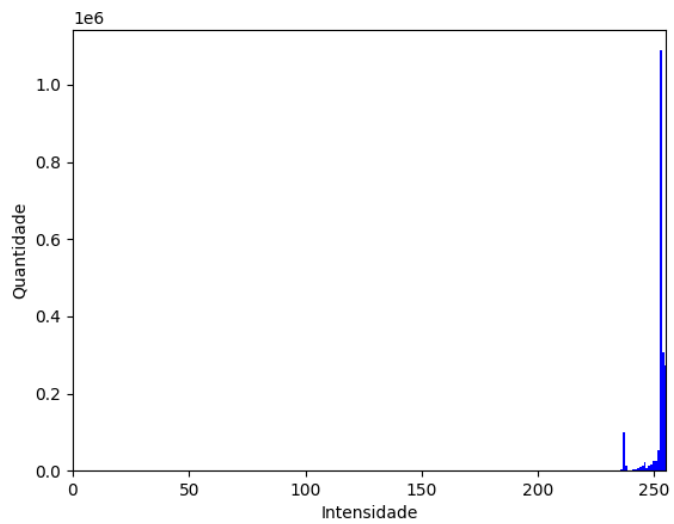
            minimo_novo_intervalo = intervalos[novo_intervalo - 1]
            maximo_novo_intervalo = intervalos[novo_intervalo]

            matriz[i][j] = minimo_novo_intervalo + ((matriz[i][j] - minimo_intervalo) / (maximo_intervalo - minimo_intervalo)) * (maximo_novo_intervalo - minimo_novo_intervalo)
```

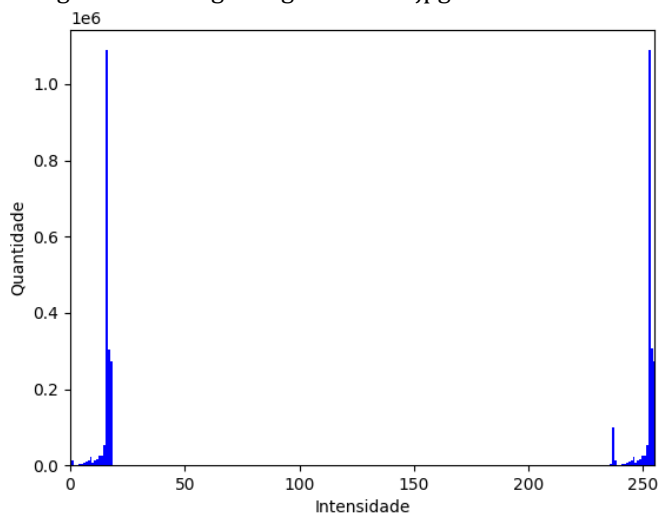
Resultados

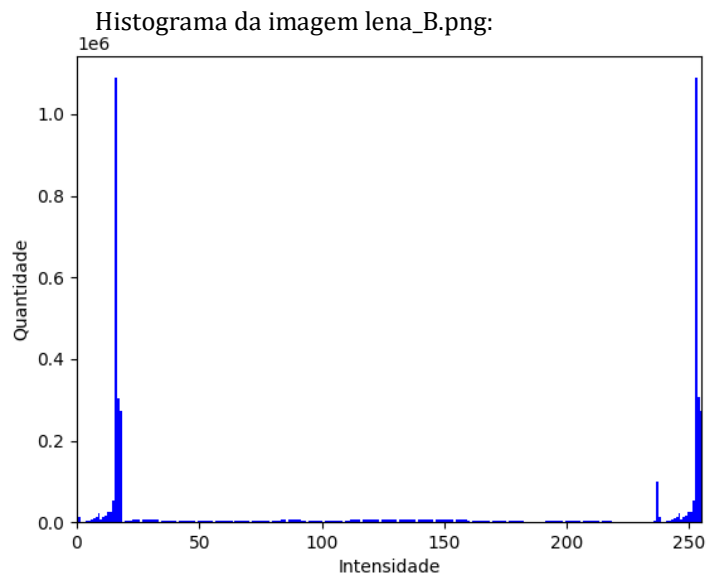
Questão 1

Histograma da imagem figuraClara.jpg:



Histograma da imagem figuraEscura.jpg:





Media, variância e entropia dos pixels das imagens:

```
Imagem figuraClara:
Média: 252.06378538951932
Variância: 17.673758398117258
Entropia: 1.5390049928081608

Imagem figuraEscura:
Média: 15.06361587097217
Variância: 17.622373495572017
Entropia: 1.5374249025941857

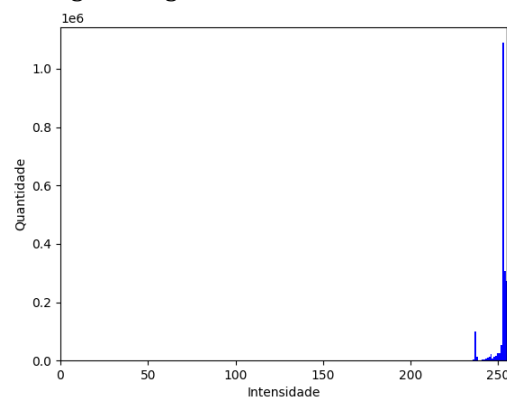
Imagem lena:
Média: 115.02948760986328
Variância: 3049.267318957427
Entropia: 5.160845421084434
```

A media de intensidade dos pixels é uma medida de brilho geral da imagem. Na imagem figuraClara, percebe-se que há uma predominância de pixels claros, enquanto na imagem figuraEscura, percebe-se uma predominância de pixels escuros, devido ao valor baixo da média dos pixels. Já na imagem lena_B, percebe-se que devido a media possuir valor próximo a 115 e a entropia possuir um valor alto, há uma alta variação de brilho, em que a imagem possui regiões mais claras e regiões mais escuras. Assim é possível perceber também que a imagem lena_B possui um alto contraste, enquanto as demais imagens possuem um baixo contraste.

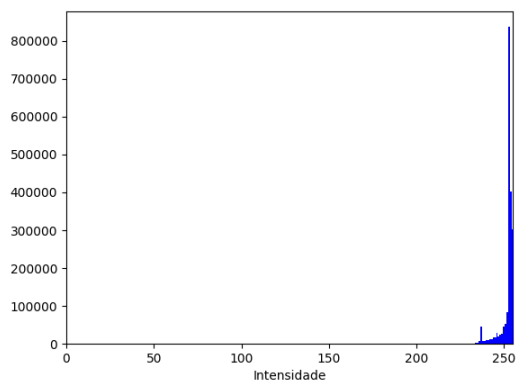
Questão 2

Histogramas antes e depois da equalização:

Histograma figuraClara antes:



Histograma figuraClara depois:

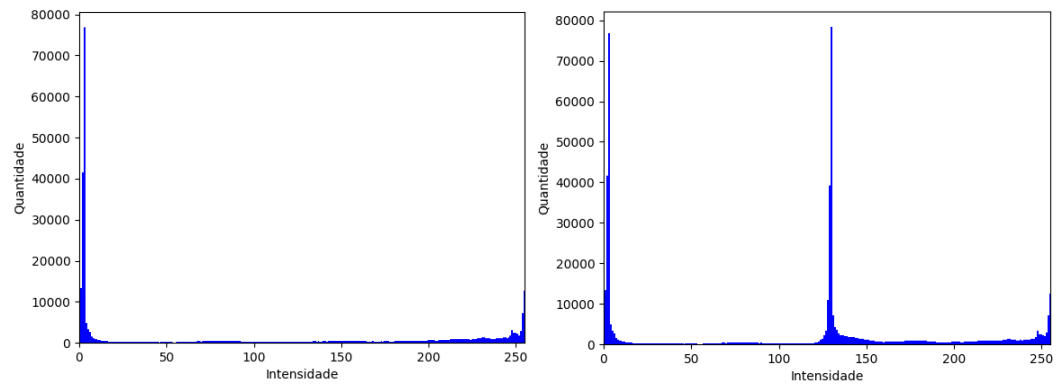


Percebe-se uma leve redistribuição no nível de intensidade dos pixels, porem apenas entre a faixa de valores que já existiam na imagem antes da equalização, observando a imagem abaixo apos a equalização percebe-se que a imagem permaneceu muito semelhante a original.

Imagem figuraClara apos equalização:

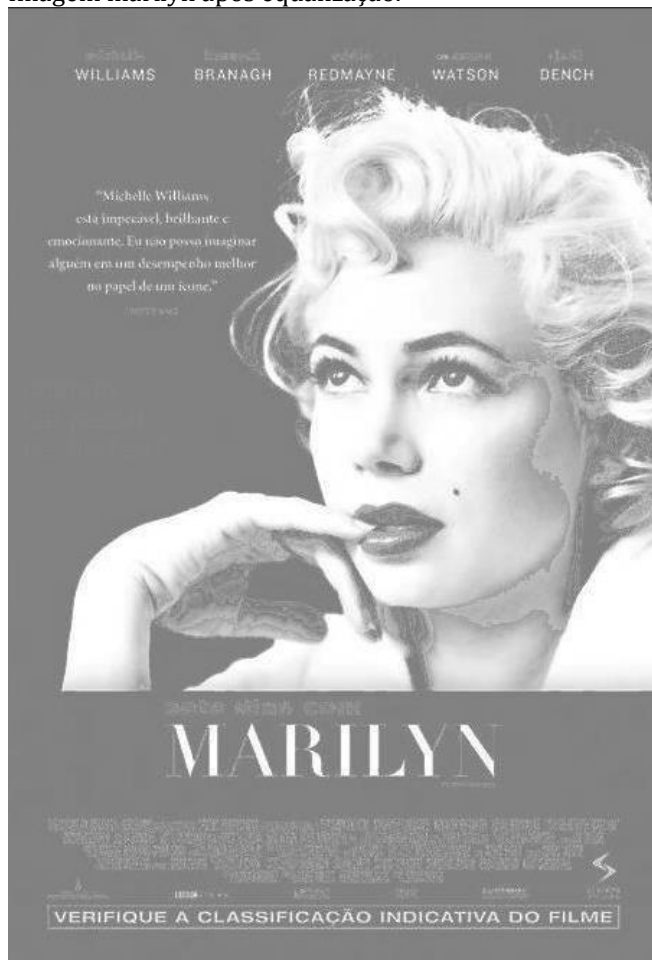
Imagem figuraEscura apos equalização:

Histograma marilyn antes: Histograma marilyn depois:



Percebe-se que houve uma redistribuição no nível de intensidade dos pixels, concentrando os valores principalmente entre 0 e 150. Assim observando a imagem equalizada abaixo, percebe-se que houve uma suavização nas cores diminuindo o contraste da imagem original.

Imagem marilyn apos equalização:



Questão 3

Resultados das imagens equalizando os canais RGB separadamente:

outono_LC:



predios:

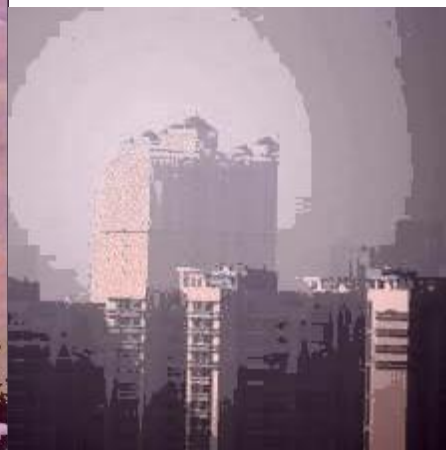


Resultados das imagens equalizando o canal Y apos converter para YIQ:

outono_LC:



predios:



Conclusões

Baseado no estudo de imagens e transformações sobre pixel, podemos concluir que o processamento de imagens é uma área de grande importância e que pode ser aplicada em diversas áreas, como medicina, engenharia, automação, entre outras. Durante o estudo, foram explorados diversos métodos para analisar e transformar as imagens, como o cálculo de média, variância, entropia e histogramas. A equalização de histograma mostrou-se uma técnica eficiente para melhorar a qualidade das imagens, melhorando o contraste e a nitidez. No entanto, também foi observado que a aplicação inadequada desta técnica pode levar a resultados indesejados, tornando a imagem artificial ou introduzindo artefatos.