**Trabalho 1**
**MO443: Introdução ao Processamento de Imagem Digital**

Henrique de Abreu Amitay - **RA:** 138501

*1 de abril de 2020*

# 1 Introdução

Este trabalho tem o objetivo de explorar alguns conceitos básicos de processamento de imagem digital e de operações em arquivos de imagens em escala de cinza.

Serão exploradas transformações lineares e não lineares para a manipulação de resolução, contrates e brilho de uma série de imagens.

Junto deste relatório se encontram os scripts para cada etapa do trabalho junto à algumas imagens de exemplo.

O trabalho foi implementado utilizando a linguagem de programação **Python** na versão **3.6.9**.

Além disso os seguintes pacotes foram utilizados para a implementação:

- OpenCv (cv2)
- Numpy
- MatplotLib

Todas as etapas deste exercício foram feitos a partir da seguinte imagem em escala de cinza:

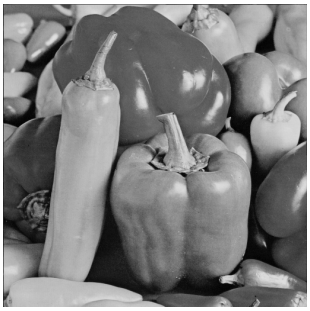

Figura 1: Imagem base utilizada em escala de cinza

# 2 Exercícios

## 2.1 Transformação de Resolução

Este procedimento consiste alterar a resolução espacial de uma imagem com as dimensões 512x512. Toda a sua implementação se encontra no arquivo **ex1.py**.

A resolução espacial de uma imagem consiste na densidade de pixels de uma imagem, ou seja, a quantidade de pixels utilizados para representar uma certa área do espaço. A resolução de uma imagem é diretamente proporcional a sua frequência de amostragem, logo, quanto menor o tempo de amostragem na captura de uma certa imagem mais pixels serão utilizados para representar uma mesma área.

O procedimento para a manipulação de resolução da imagem acima consistiu em redimensiona-la de forma a diminui-la e aumenta-la ao seu tamanho original para que possamos observar o efeito na resolução da imagem.

Ao diminuir as dimensões de uma imagem, temos que grupos de pixels são "comprimidos"em um único pixel de forma, logo existe uma perda de informação neste processo. Ao recuperar as dimensões da imagem, não existem mais dados que nos possibilitam recuperar os pixels da área reduzida, logo populamos está area com o mesmo valor de intensidade do pixel reduzido.

Se definirmos processo de redução pode ser definido como uma função:

$$f(A) = p \tag{1}$$

Aonde $A$ é o vetor de pixels a serem reduzidos e $p$ é a intensidade resultante do processo de redução. Mesmo que saibamos a função, existem múltiplas elementos de seu dominio que resultam no elemento $p$ da imagem.

Podemos concluir então que ao diminuir a resolução de uma imagem existe uma perda irreversivel de informação.

O resultado do procedimento acima para a imagem da Figura 1, se encontra a seguir:

Figura 2: Imagem com resolução reduzida para 128x128

Figura 3: Imagem com resolução reduzida para 64x64

Figura 4: Imagem com resolução reduzida para 32x32

Figura 5: Imagem com resolução reduzida para 16x16

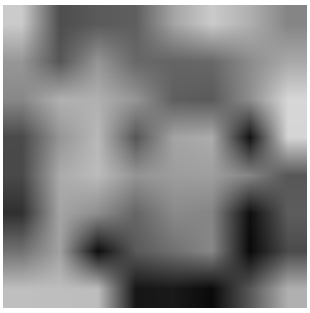

Figura 6: Imagem com resolução reduzida para 8x8

## 2.2 Quantização de Imagens

Este procedimento consiste em alterar os níveis de quantização de uma imagem. Toda a sua implementação se encontra no arquivo **ex2.py**.

A quantização refere-se ao número de níveis de cinza utilizados para representar uma imagem monocromática, ela é fortemente relacionada a profundidade de uma imagem, ou seja, o número de bits necessários para a sua armazenação.

Para este procedimento, uma imagem com 256 níveis de cinza teve sua quantização alterada a partir de uma operação lógica que consistiu em remover o bit menos significativo de sua representação binária gradativamente.

Para isto foi utilizada uma máscara de bits para cada nível desejado e a operação lógica `AND`, conforme trecho de código:

```
# Mascara usada para quantizar a imagem para 16
    niveis
mask = 0b11110000
img = img & mask

```

Este procedimento também escurece a imagem, dado que a operação lógica gradativamente diminui as intensidades que ainda não foram zeradas com a operação, logo para facilitar na visualização foi feita uma transformação arbitrária em que o maior valor de intensidade resultante de cada imagem após a quantização teve seu valor alterado para branco (255).

O resultado das quantizações se encontram a seguir:

Figura 7: Imagem com profundidade de 64 níveis

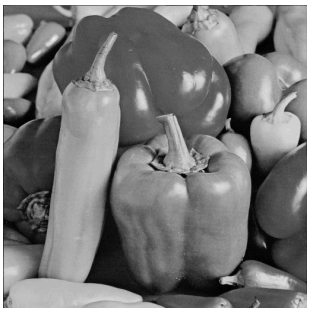

Figura 8: Imagem com profundidade de 32 níveis

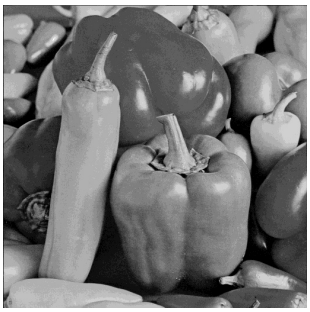

Figura 9: Imagem com profundidade de 16 níveis

Figura 10: Imagem com profundidade de 8 níveis

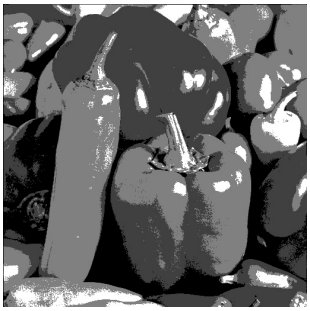

Figura 11: Imagem com profundidade de 4 níveis

Figura 12: Imagem com profundidade de 2 níveis

## 2.3 Escala de Cinza

Este procedimento consiste em explorar diferentes transformações lineares e não lineares e os seus efeitos no brilho e contraste da imagem. Toda a sua implementação se encontra no arquivo **ex3.py**.

Para fins comparativos, além da imagem, iremos explorar o histograma de níveis de cinza da imagem de forma a nos fornecer subsidios para melhor entender o efeito de cada transformação.

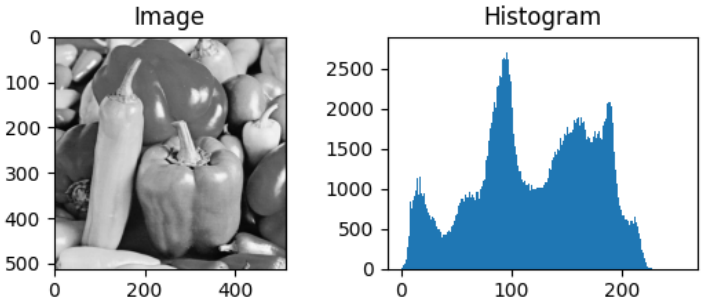


Figura 13: Histograma da imagem sem nenhuma transformação

### 2.3.1 Transformação pelo logaritmo

A transformação pelo logaritmo consiste em aplicar a seguinte função em cada pixel:

$$g(f) = alog(f + 1) \tag{2}$$

Aonde $a$ é um valor arbitrário, em nosso caso definido para $a = \frac{255}{log(1+f_{max})}$, com $f_{max}$ sendo a maior intensidade encontrada na imagem. Este valor de $a$ garante que as intensidades dos pixels após a transformação se encontrarão na escala $[0, 255]$

Após executar a transformação, obtemos a seguinte imagem:

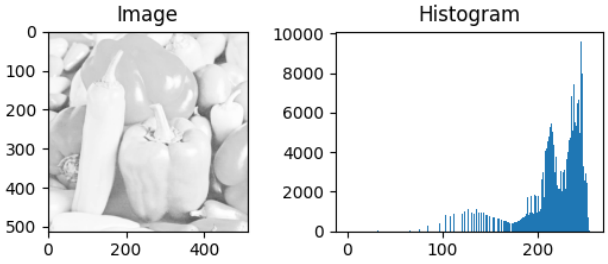


Figura 14: Histograma da imagem após transformação pelo logaritmo

Podemos perceber pela imagem que o seu brilho aumentou consideravelmente, porém houve um encurtamento de seu contraste. Esta observação é corroborada pelo histograma, no qual podemos ver que a frequência dos pixels

se concentra nas maiores intensidades. Podemos concluir então que esta transformação possui um realce maior em pixels mais escuros, de baixa intensidade.

### 2.3.2 Transformação pelo exponencial

A transformação pelo exponencial consiste em aplicar a seguinte função em cada pixel:

$$g(f) = a(\exp f - 1) \tag{3}$$

Aonde $a$ é um valor arbitrário, em nosso caso definido para $a = \frac{255}{\exp 255 - 1}$. Este valor de $a$ garante que as intensidades dos pixels após a transformação se encontrarão na escala $[0, 255]$

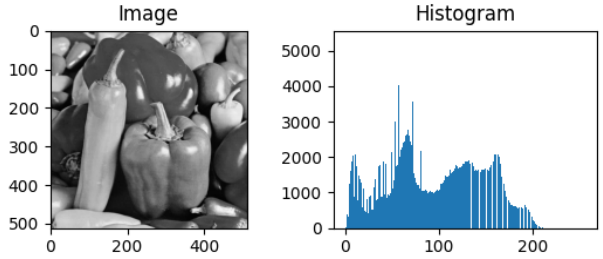


Figura 15: Histograma da imagem após transformação pelo exponencial

Podemos observar que a imagem teve seu brilho reduzido consideravelmente, devido ao realce focado nos pixels de maior intensidade. A imagem também teve seu contraste reduzido consideravelmente dado que a distribuição das intensidades se concentrou nas bandas de menor intensidade.

### 2.3.3 Transformação pelo quadrado

A transformação pelo quadrado consiste em aplicar a seguinte função em cada pixel:

$$g(f) = a(f^2) \tag{4}$$

Aonde $a$ é um valor arbitrário, em nosso caso definido para $a = \frac{1}{255}$, com $f_{max}$. Este valor de $a$ garante que as intensidades dos pixels após a transformação se encontrarão na escala $[0, 255]$

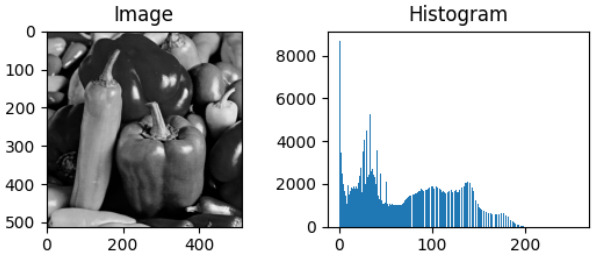


Figura 16: Histograma da imagem após transformação pelo quadrado

Podemos observar que a imagem teve seu brilho reduzido consideravelmente, porém não na mesma intensidade que a transformação exponencial. Este escurecimento mais brando manteve mais parte do contraste da imagem.

### 2.3.4 Transformação pela raiz quadrada

A transformação pela raiz quadrada consiste em aplicar a seguinte função em cada pixel:

$$g(f) = a\sqrt{f} \tag{5}$$

Aonde $a$ é um valor arbitrário, em nosso caso definido para $a = \frac{255}{\sqrt{255}}$.. Este valor de $a$ garante que as intensidades dos pixels após a transformação se encontrarão na escala $[0, 255]$

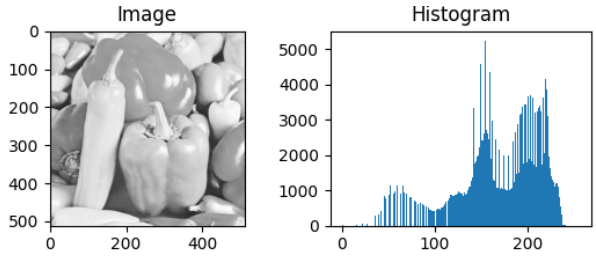


Figura 17: Histograma da imagem após transformação pela raiz quadrada

Podemos perceber pela imagem que assim como a transformação pelo logaritmo o seu brilho aumentou consideravelmente e também houve um encurtamento do contraste. Ambos estes efeitos ocorreram com uma intensidade menor que o logaritmo, o que é esperado dado a diferença entre ambas as curvas.

O histograma nos fornece as mesmas informações, porém com um efeito que também contempla os pixels de média intensidade, garantindo um contraste maior que a transformação pelo logaritmo mas ainda assim menor que a imagem original.

### 2.3.5 Alargamento de contraste

O alargamento de contraste consiste na seguinte operação:

$$g(f) = \begin{cases} \alpha f & \text{se } 0 \leq \text{f} \leq \text{a} \\ \beta(f-a) + \alpha a & \text{se a} \leq \text{f} \leq \text{b} \\ \gamma(f-b) + \beta(b-a) + \alpha a & \text{se b} \leq \text{f} \leq \text{L} \end{cases} \quad (6)$$

Aonde $a$ e $b$ são valores de uma banda de intensidades, nesse caso escolhemos os maiores picos a partir do histograma inicial. E $\alpha$, $\beta$ e $\gamma$ são os valores dos coeficientes angulares das seguintes retas:

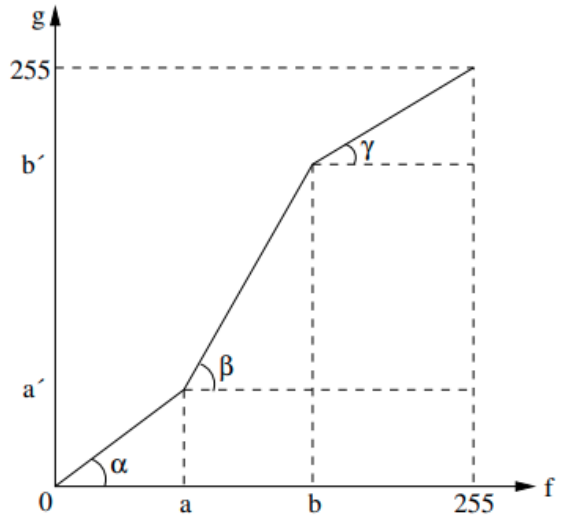


Figura 18: Gráfico representando a transformação de alargamento de contraste

Com os valores, escolhidos a transformação pode ser aplicada de forma vetorizada a partir da vetorização de funções proporcionadas pelo `numpy`, conforme o exemplo de código:

```
def enlarge_contrast(img):
    ximg = img.copy()

    # Arbitrary values chosen based on the
    original histogram
    a = 74
    alpha = 0.5

    b = 163
    beta = 1.73

    gamma = 0.7

    def contrast_func(pixel):
        if (pixel >= 0 and pixel <= a):
            return alpha*pixel

        if (pixel > a and pixel <= b):
            return beta*(pixel - a) + alpha*a

        if (pixel > b and pixel <= MAX_LEVEL):
            return gamma*(pixel - b) + beta*(b -
    a) + alpha*a

        return pixel

    vfunc = np.vectorize(contrast_func)
    return vfunc(img)
```

O resultado do alargamento de contraste se encontra na figura a seguir:

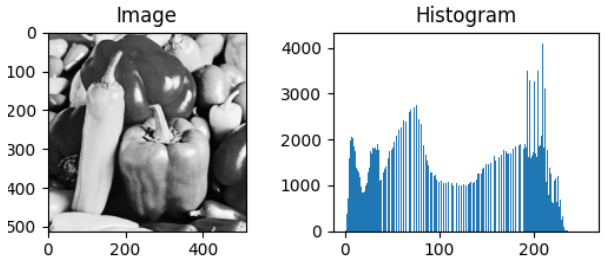


Figura 19: Histograma da imagem após transformação de alargamento de contraste

Podemos ver que após a transformação o histograma ficou mais largo e com menos picos. Além disso fica bem claro a influência de $\beta$ em distribuir os valores do meio da banda para as extremidades.