

Universidade de São Paulo



## **Trabalho 2**

# Análise de Histogramas para a Aplicação de Transformações Lineares e Máscaras

Processamento e Análise de Imagens

**Aluno**

Thiago Oliveira dos Santos n° 13696220

São Carlos  
2024

# 1 Objetivos

Este trabalho tem como objetivo aplicar os conceitos de histogramas, transformações lineares e máscaras em imagens digitais. Serão exploradas técnicas como visualização de histogramas, normalização de histogramas, escurecimento e redução de contraste de imagens, equalização de histogramas e aplicação de máscaras binárias para selecionar regiões específicas de uma imagem.

## 2 Introdução

### 2.1 Histogramas

Um histograma de uma imagem digital é uma representação gráfica da distribuição de intensidades de pixels na imagem. Ele é útil para entender a distribuição de brilho e contraste na imagem; daí, permitindo análises e possíveis manipulações que possam ser feitas na imagem. Por exemplo, um histograma com acúmulo de frequências na direita implicaria numa imagem muito clara; por conseguinte, uma transformação linear poderia ser aplicada para que a imagem fosse escurecida.

#### 2.1.1 Normalização de um histograma

A normalização de um histograma é o processo de ajustar os valores do histograma para um intervalo de 0 a 1. Isso consiste em pegar cada valor de frequência do histograma, e dividir pela quantidade de pixels  $M \times N$  da imagem. Um histograma assim, caso seja escolhida uma escala adequada no eixo das frequências tem exatamente o mesmo formato que o histograma original, e pode ser interpretado como a probabilidade de, escolhendo um pixel aleatório da imagem, ele ter certa intensidade ou não. Logo, uma das propriedades mais notáveis de um histograma assim, é que a soma total de cada uma das barras q faz o histograma resulta em 1.

#### 2.1.2 Contraste

O contraste em uma imagem se refere à diferença entre as intensidades dos pixels. O histograma de uma imagem pode facilmente indicar se uma imagem tem alto ou baixo contraste, o que influencia na sua percepção visual. Um exemplo seria se o histograma consistisse de "dois morros separados por um vale", que indicaria a existência de um alto contraste.

### 2.2 Transformações lineares

Sabe-se que uma imagem pode ser representada por uma função  $f(x, y)$ , onde cada pixel  $(x, y)$  retorna uma intensidade  $f$  da imagem. Uma transformação linear  $T(f)$  em uma imagem, é uma operação que mapeia os valores dos pixels para novos valores usando uma função linear. Ou seja, a aplicação de  $T$  criaria uma nova imagem  $g(x, y)$ ; de forma tal que  $g$  seria dada pela Equação 1.

$$g(x, y) = af(x, y) + b = T(f) \quad (1)$$

Aqui pode não ser claro a uma primeira vista, mas afirmo (e nas próximas seções o leitor poderá verificar) que  $a$  representa uma mudança no contraste de  $f$ , e  $b$

representa uma mudança em seu brilho. Dito isso, destaco que transformações lineares são ferramentas simples, mas bem úteis quando se pretende mexer no brilho e contraste de uma imagem.

### 2.2.1 Equalização de um histograma

A equalização de um histograma é uma técnica usada para uniformizar a distribuição de intensidades de imagens. Seu intuito é tornar um histograma mais uniforme; e, geralmente, isso é feito pela aplicação de, nada mais nada menos que, uma **transformação linear**.

## 2.3 Máscaras binárias

Uma máscara binária para imagens digitais é uma matriz binária que seleciona determinadas regiões da imagem com base em critérios específicos. Esta pode ser usada para aplicar efeitos locais ou para segmentar objetos em uma imagem. No geral, gosto de pensar na aplicação de uma máscara como uma transformação não linear  $R(f, m)$  que retorna uma imagem  $g(x, y)$ , onde  $f(x, y)$  representa a imagem, e  $m(x, y)$  representa a máscara. Não necessariamente a máscara deve representar uma imagem por si só, uma vez que para cada  $(x, y)$  da imagem,  $0 \leq m(x, y) \leq 1$ . A aplicação  $R$  é dada, então, pela Equação 2.

$$g(x, y) = f(x, y) \cdot m(x, y) = R(f, m) \quad (2)$$

Ou seja, cada intensidade em uma posição  $(x, y)$  da imagem é multiplicada por 0 ou 1, a depender do que está em  $m(x, y)$ . Nota-se, então, que a máscara é usada justamente para "mascarar" certas partes da imagem, uma vez que essa escurece as posições  $(x, y)$  da imagem que correspondem a  $m(x, y) = 0$ .

## 3 Metodologia e Resultados

### 3.1 Materiais anteriores

Antes de abordar os métodos utilizados para resolver os problemas propostos neste trabalho, é importante ressaltar que estou continuando o desenvolvimento da biblioteca própria de manipulação de imagens apresentada no trabalho anterior. Neste trabalho, foquei em aprimorar essa biblioteca, adicionando novos métodos e funcionalidades.

Entre as adições feitas, destacam-se:

- **Novos Métodos Adicionados:** Foram adicionados os métodos *transformacao\_linear*, *aplica\_mascara* e *mascara\_bin* à classe *Image*. Esses métodos proporcionam novas maneiras de manipular e transformar imagens dentro da biblioteca.
- **Nova Função Adicionada:** Introduzimos a função *plota\_historama*, que permite plotar o histograma de uma imagem. Essa função substituiu a anterior *plota\_perfil\_linha*, embora mantenha a funcionalidade desta última, uma vez que o método *perfil\_linha* passou a ser apenas uma função que retorna um histograma com o perfil de uma linha.

- **Importação de Novo Módulo:** Para dar suporte à função *aplica\_mascara*, importamos a função *deepcopy* do módulo *copy*. Isso permite a cópia profunda de objetos, o que é essencial para evitar alterações indesejadas na imagem original durante o processo de aplicação de máscara.
- **Nova Função de Normalização:** Foi adicionada a função *normaliza\_hist*, que normaliza o histograma de uma imagem. Essa função é útil para análise estatística e visualização de distribuições de intensidade de pixel.

Essas adições fortaleceram a funcionalidade da biblioteca de manipulação de imagens, tornando-a mais versátil e poderosa para lidar com uma variedade de tarefas de processamento de imagem. Com isso, foi possível resolver todos os problemas propostos. Segue em anexo a nova versão da biblioteca: ➡

## 3.2 Visualização do histograma

Para as seguintes seções, video o arquivo do trabalho: ➡

Nesse trabalho foram usadas como objeto de estudos as Imagens 1 e 2.

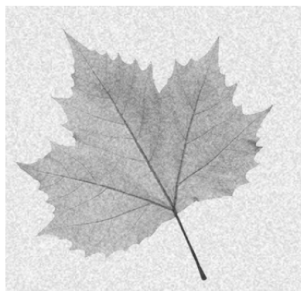


Figura 1: Imagem de um folha com alto brilho

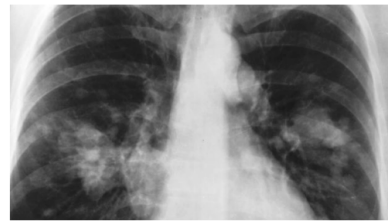


Figura 2: Imagem de um tórax com alto contraste

Para visualizar o histograma das imagens, foi utilizada a função *plota\_histograma* da biblioteca. Essa função exibe o histograma das imagens em questão; que, dentro da classe *Image*, já possuem o atributo *.hist*, que é seu histograma próprio. Os histogramas estão representados nas Imagens 3 e 4.

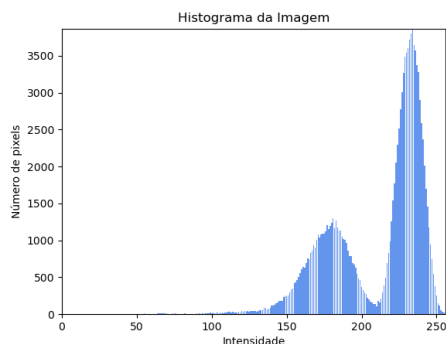


Figura 3: Histograma da Imagem 10

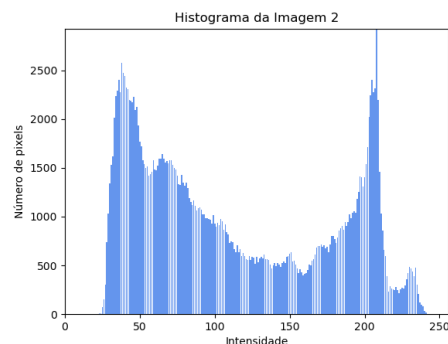


Figura 4: Histograma da Imagem 11

### 3.3 Normalização do histograma

Para esta seção, foi usada a Imagem 1. O histograma normalizado está apresentado na Imagem 5.

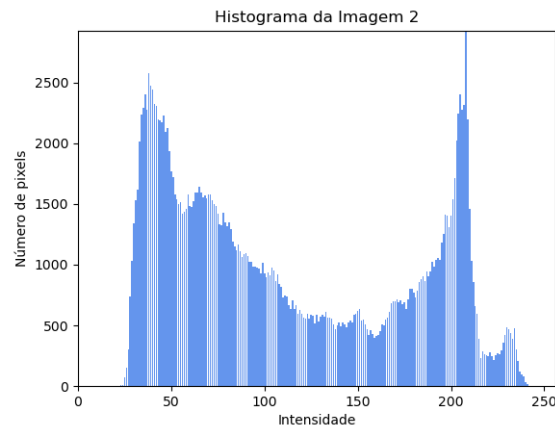


Figura 5: Histograma Normalizado

Note que este tem a mesma cara que o histograma da Imagem 3, a despeito apenas da escala do eixo y. Note também como a escala agora está no intervalo de 0 a 1, donde podemos tirar a interpretação de porcentagens de distribuições de pixels na imagem.

### 3.4 Escurecer uma imagem

Para esta e as próximas seções em que se faz uso de transformações lineares, utilizei o método *transformacao\_linear* da biblioteca. Analisando os histogramas das Imagens 1 e 2, decidi por trabalhar com a imagem 1 nesta seção, uma vez que essa apresenta um histograma mais concentrado na região com altas intensidades, o que implica numa imagem clara. A imagem escurecida e seu histograma estão representados nas Imagens 6 e 7, respectivamente.



Figura 6: Imagem da Folha Escurecida

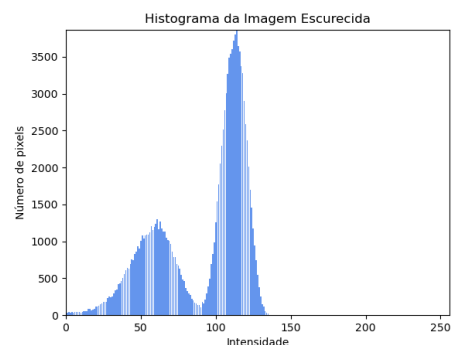


Figura 7: Histograma da Imagem Escurecida

Para a aplicação da transformação linear, foram escolhidos os parâmetros 1 e  $-120$  como contraste e brilho, respectivamente (vide Equação 1). Note como essa transformação apenas transladou o histograma original para a esquerda, consequentemente escurecendo a imagem, mas ainda mantendo seus detalhes.

### 3.5 Diminuir o contraste de uma imagem

Analizando os histogramas das Imagens 1 e 2, decidi por trabalhar com a imagem 2 nesta seção, uma vez que essa apresenta um histograma mais separado: possui uma concentração na região mais clara, e outra na região mais escura, o que implica numa imagem com contraste maior. A imagem com menos contraste e seu histograma estão representados nas Imagens 8 e 9, respectivamente.

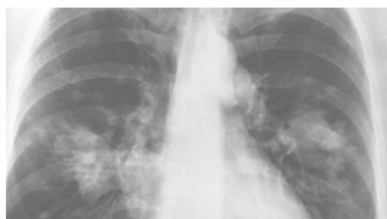


Figura 8: Imagem do Tórax com Menos Contraste

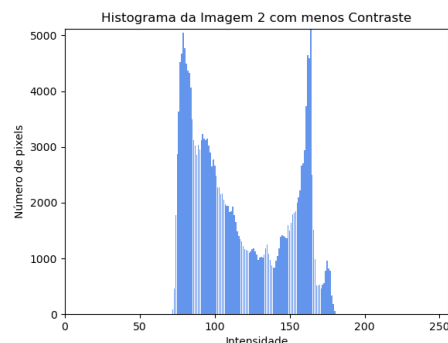


Figura 9: Histograma da Imagem com Menos Contraste

Para a aplicação da transformação linear, foram escolhidos os parâmetros 0.5 e 60 como contraste e brilho, respectivamente. Note como essa transformação contraiu o histograma original para o centro, conseqüentemente diminuindo o contraste da imagem. Isso apenas comprova ainda mais o que havia sido dito a respeito de o contraste ser indicado pela concentração de frequências nas extremidades do histograma: quanto mais longes essas extremidades do centro, mais contraste tem a imagem.

### 3.6 Equalização do histograma

Para esta seção, sem perda de generalidade, decidi trabalhar com ambas as imagens. As imagens equalizadas estão representadas nas Imagens 10 e 11, enquanto seus respectivos histogramas estão referenciados nas Imagens 12 e 13.

Para as equalizações, foram realizadas transformações lineares também. Os parâmetros da primeira são 2.2 e 300; e os parâmetros da segunda são 1.25 e  $-30$ . Como início da discussão, gostaria de voltar a atenção sobre como, de certa forma, as Imagens 10 e 6 se parecem. Isso pode ser devido ao fato de a imagem original ser excessivamente clara; e, de certa forma, ambas as transformações lineares aumentarem o contraste da imagem. Note também como o contraste da imagem do tórax aumentou mesmo em relação à original. Analisando o histograma de ambas as imagens, isso se dá porque a equalização nada mais é que a tentativa do melhor fitting para histograma, onde ele fica tão reto quanto possível. No geral, é possível perceber como a equalização aprimora o contraste, e deixa os detalhes mais nítidos.

### 3.7 Mascaram o fundo de uma imagem

Desenvolvi uma função na biblioteca para criar máscaras binárias e segmentar (mascarar) regiões de interesse. Primeiramente, escolhi a imagem da folha para mascarar um fundo, uma vez que essa tem um fundo muito bem definido. Daí,



Figura 10: Imagem da Folha Equalizada



Figura 11: Imagem do Tórax Equalizada

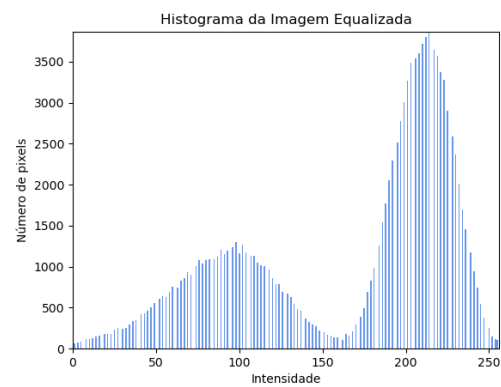


Figura 12: Histograma da Folha Equalizada

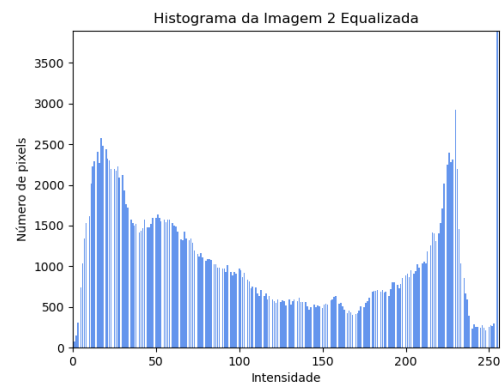


Figura 13: Histograma do Tórax Equalizado

analisando o histograma, notei que o fundo correspondia, em sua devastadora maioria, de pixels com intensidade maior ou igual a 210. Daí, a partir do algoritmo para a criação da máscara binária e sua aplicação, representado no Código 1, foi feita a Imagem 14 junto de seu histograma na Imagem 15.

Listing 1: Aplicação de máscara binária

```
def aplica_mascara(self, outputpath, mascara):
    with open(outputpath, "w") as arquivo:
        arquivo.write(self.magic + "\n") #escreve o numero magico
        arquivo.write(str(self.ncoluna) + "-" + str(self.nlinha) + "\n") #escreve a dimensao da imagem
        imagem = deepcopy(self.pixels) #inicializa os pixels da nova imagem

        for i in range(self.nlinha):
            for j in range(self.ncoluna):
                imagem[i][j] *= mascara[i][j] #coloca a mascara por cima da imagem

        maximo_imagem = self._max_matriz(imagem)
        arquivo.write(str(maximo_imagem) + "\n")

        self._escreve_pixels(arquivo, imagem)

    return Image(outputpath)

def mascara_bin(self, limite, maior = True): #cria uma mascara binaria
    mascara = [] #inicializa a mascara

    if maior: #caso se queira mascarar valores maiores que limite
        for i in self.pixels:
            linha = []
            for j in i:
                if j > limite: #caso seja um pixel que sera mascarado
                    linha.append(0)
                else:
                    linha.append(1)
            mascara.append(linha)

    else: #caso se queira mascarar valores menores que limite
        for i in self.pixels:
            linha = []
            for j in i:
                if j < limite: #caso seja um pixel que sera mascarado
                    linha.append(0)
                else:
                    linha.append(1)
            mascara.append(linha)

    return mascara
```

O algoritmo basicamente consiste em ver se os pixels da imagem são maiores que o valor especificado; e, caso sejam, a máscara terá um valor de 0 na posição correspondente. Consequentemente, por esse algoritmo, como pode-se ver na



Imagem 14, o fundo foi "mascarado"(zerado). Analisando o histograma, pela escala, nota-se um valor tremendo no y, o que é de se esperar, uma vez que grande parte dos pixels da imagem foram zerados. O que não foi zerado (a folha), continua com o histograma normal, apenas numa escala diferente.



Figura 14: Imagem após a Máscara

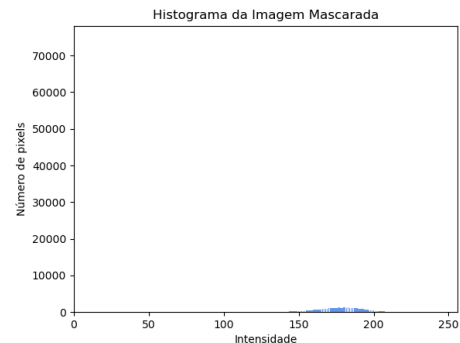


Figura 15: Histograma da Imagem Mascarada

## 4 Conclusão

Neste trabalho, exploramos diversas técnicas de processamento de imagens, incluindo análise de histogramas, aplicação de transformações lineares e uso de máscaras. Essas técnicas são fundamentais para ajustar o brilho, contraste e segmentação de imagens digitais. Através da aplicação dessas técnicas em uma imagem de exemplo, pudemos observar como cada uma afeta a distribuição de intensidades dos pixels e a percepção visual da imagem. Este trabalho serviu como uma introdução prática ao processamento de imagens e ofereceu insights importantes para análises futuras.