

Trabalho 2 - MC920

Rian Radeck Santos Costa - 187793

27 de Setembro de 2022

Sumário

1	Especificações do programa	3
2	Algoritmos	3
3	Método Global	5
4	Método de Bernsen	6
5	Método de Niblack	7
6	Método de Sauvola e Pietaksinen	8
7	Método de Phansalskar, More e Sabale	10
8	Métodos de Contraste, Média e Mediana	11
9	Histogramas	13
9.1	Fiducial	13
9.2	Peppers	14
9.3	Sonnet	15
9.4	Baboon	16
9.5	Monarch	17
9.6	Retina	18
9.7	Wedge	19

1 Especificações do programa

Todas as imagens na pasta desse trabalho (./binary_images) foram geradas com vizinhança 5 x 5 e os valores padrões sugeridos, exceto quando existe um arquivo de texto com os valores utilizados.

Notei algumas coisas sobre esse trabalho:

- Valores altos (entre 15 e 50) para vizinhança geralmente funcionam melhor.
- Os valores de k são como uma tolerância para bordas e ruídos.
- O método de Bernsen é o mesmo que o método do contraste com objeto e fundo invertidos.
- Foi pedido que fossem gerados histogramas para as imagens binárias, porém isso nos daria a mesma informação da porcentagem de preto delas, sendo assim fiz os histogramas das imagens originais, o que nos fornece a informação do contraste das imagens.
- Os k 's não deveriam ser negativos para que o “objeto” da imagem (e.g. texto) continue preto?

Encorajo fortemente também que o código seja executado para uma melhor visualização das imagens e um melhor entendimento sobre os parâmetros dos métodos apresentados, principalmente dos métodos mais complexos e da aproximação contínua dos histogramas. Enfim, esclarecidos os passos iniciais vou mostrar as imagens que melhor se encaixaram em cada método e suas porcentagens de preto.

2 Algoritmos

Todos os métodos foram implementados de maneira similar, mudando somente a maneira que é calculado o limiar para cada pixel.

A implementação ficou bem simples: para cada pixel era considerada uma vizinhança de $n \times n$ centrada nele, feita com slicing em python. Caso o slicing caísse fora dos limites de altura e largura da imagem, seus valores eram “clipados” para os limites, ou seja, nem sempre a vizinhança era $n \times n$, por exemplo para o pixel (0,0) o tamanho de sua vizinhança era aproximadamente $\frac{1}{4}$ do tamanho da vizinhança de pixels mais centralizados. Algo

também a ser notado é que quando n é par, ele é arredondado para cima, por exemplo uma vizinhança 2 x 2 é na verdade uma vizinhança 3 x 3.

Segue um pseudocódigo da implementação desses algoritmos:

Algorithm 1 Generic Algorithm (img, n)

```

new_img  $\leftarrow$  img
for  $i \leftarrow 0 \dots img.height-1$  do
  for  $j \leftarrow 0 \dots img.width-1$  do
    start_row  $\leftarrow \max(0, i - \lfloor \frac{n}{2} \rfloor)$ 
    end_row  $\leftarrow \min(img.height, i + \lfloor \frac{n}{2} \rfloor)$ 
    start_column  $\leftarrow \max(0, j - \lfloor \frac{n}{2} \rfloor)$ 
    end_column  $\leftarrow \min(img.width, j + \lfloor \frac{n}{2} \rfloor)$ 

    neighbourhood  $\leftarrow img[start\_row \dots end\_row][start\_column \dots end\_column]$ 
    threshold  $\leftarrow$  GivenMethod(neighbourhood)
    if  $img[i][j] > threshold$  then
      new_img[i][j]  $\leftarrow$  Object
    else
      new_img[i][j]  $\leftarrow$  Background
    end if
  end for
end for
return new_img

```

Os cálculos feitos pela função GivenMethod foram os fornecidos no enunciado do trabalho.

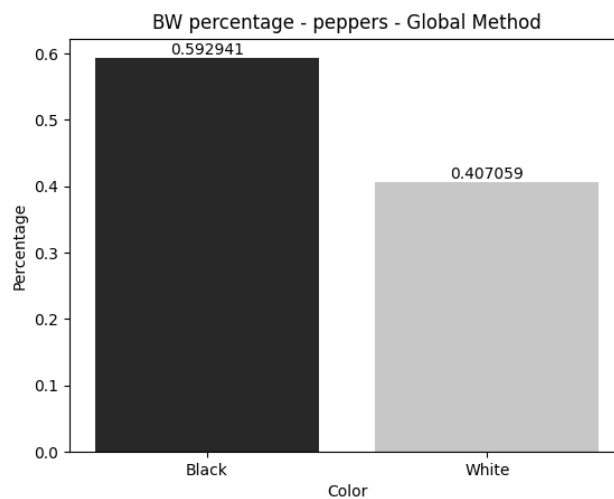
OBS: Note que no método de Phansalskar, More e Sabale, os valores são divididos por 255 (normalizados) antes que seja feito o cálculo do limiar.

3 Método Global

Considerando a simplicidade e velocidade de execução desse método (não considera vizinhança), acho que ele melhor se encaixa mostrando onde se encaixam as sombras de uma imagem. Observe na imagem das pimentas



Método Global ($T = 100$)

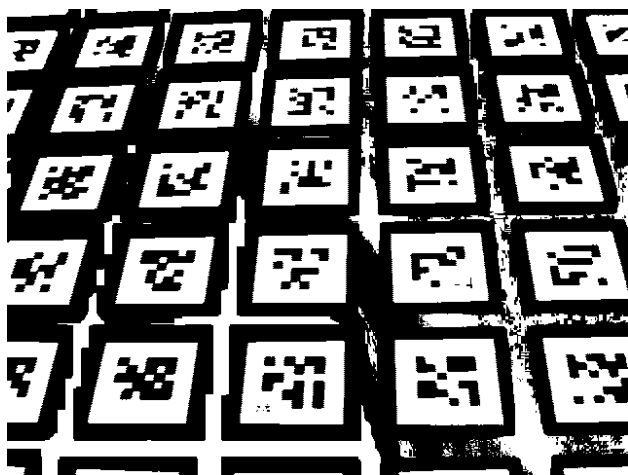


Porcentagem da coloração da imagem binária

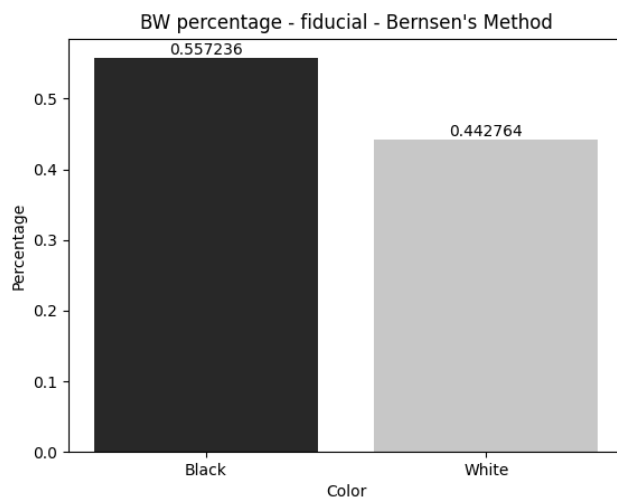
Veja que é bem claro onde estão as sombras da imagem, ainda mais que essa é uma imagem de contraste razoável, como é possível ver em 9.2.

4 Método de Bernsen

Gostei bastante do resultado obtido no Fiducial com $n = 25$. Os quadradinhos ficaram bem separados e definidos dada a simplicidade do método, além de que a parte escura da imagem foi bem processada.



Método de Bernsen ($n = 25$)

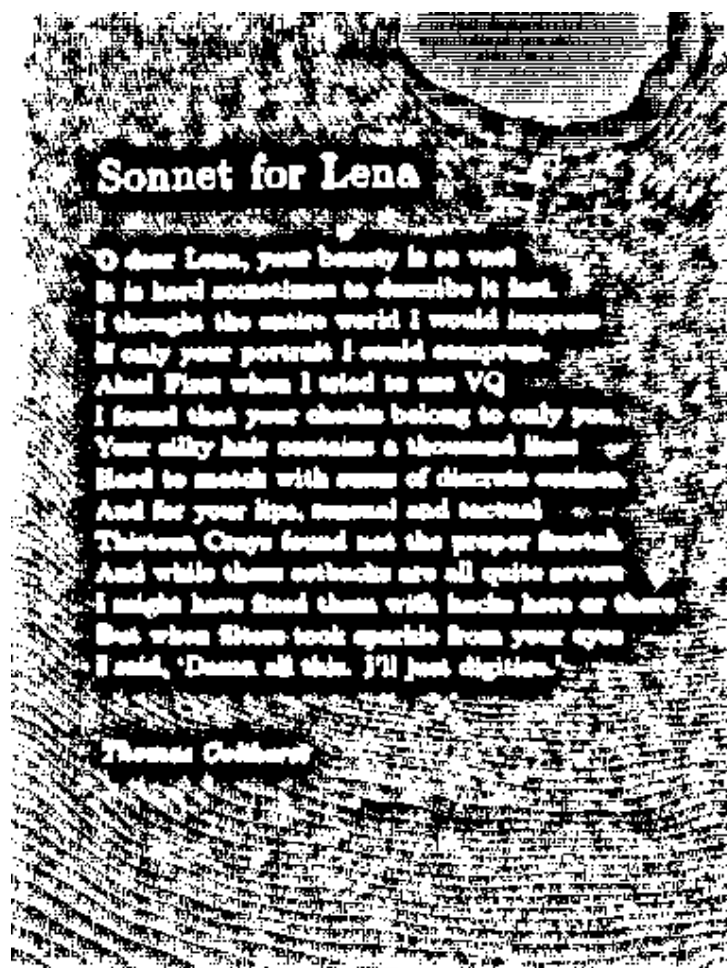


Porcentagem da coloração da imagem binária

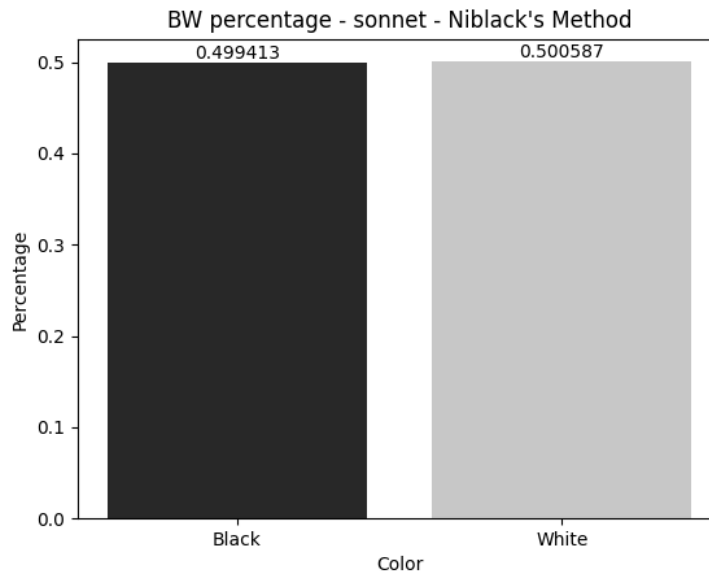
Claramente, o método do Contraste fornecerá um resultado similar, o que faz sentido, já que essa imagem tem um baixo-médio nível de contraste, como podemos ver em 9.1.

5 Método de Niblack

É um método muito interessante, já um pouco mais complexo. Notei que em áreas de cor uniforme existia bastante ruído, dessa forma fui observando que era uma boa maneira de identificar fontes de luz, o que fica bem evidente no exemplo a seguir



Método de Niblack ($n = 25; k = 0, 1$)



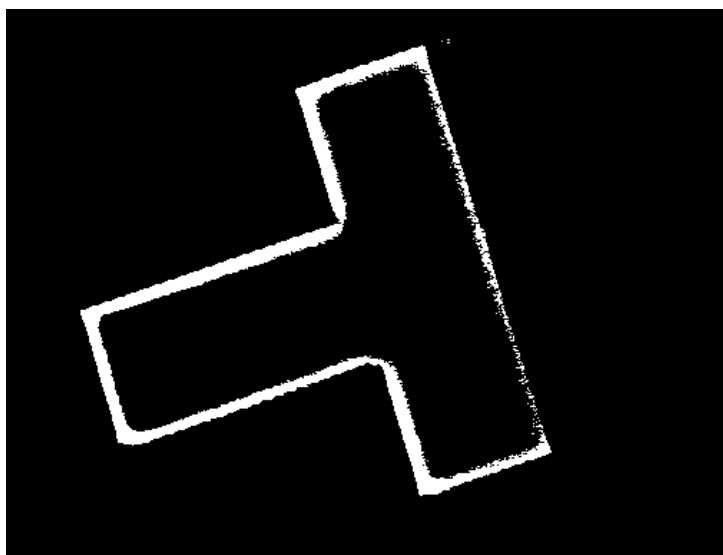
Porcentagem da coloração da imagem binária

Claro que isso é só uma observação menor no método de Niblack; esse método é na verdade um dos métodos pioneiro na binarização de imagens e funciona muito bem em textos e imagens onde o fundo não é uniforme em geral.

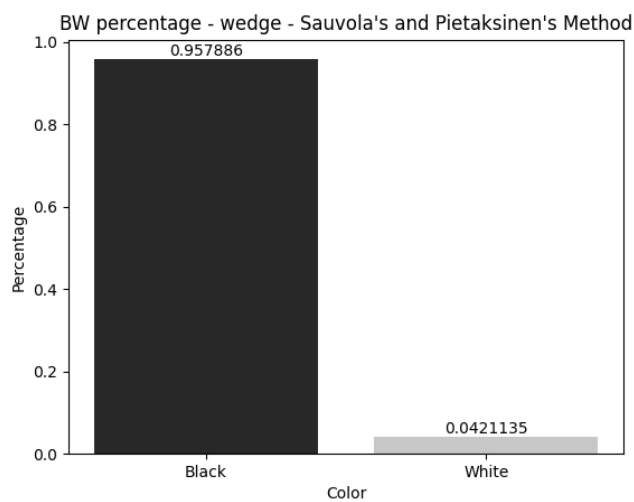
Esse método também mostra um resultado bem interessante com o wegde, acredito que uma imagem de bom contraste seja fundamental já que uma propagação de luz cria um degradê de contraste, podemos observar o histograma de contraste do sonnet em 9.3.

6 Método de Sauvola e Pietaksinen

Realmente um método mais complexo e adaptável para binarização de imagens, aqui é possível obter filtragens bem precisas de diversos fatores da imagem (e.g. bordas), como é possível observar na seguinte imagem



Método de Sauvola e Pietaksinen ($n = 27$; $k = 0, 12$; $R = 72$)

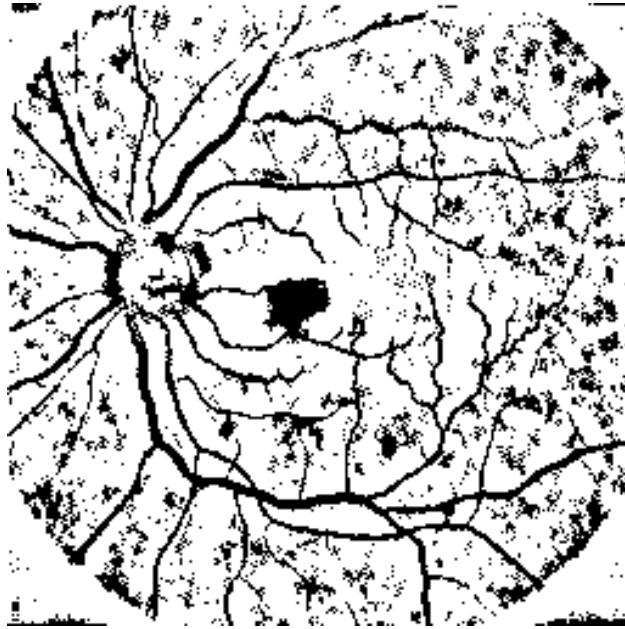


Porcentagem da coloração da imagem binária

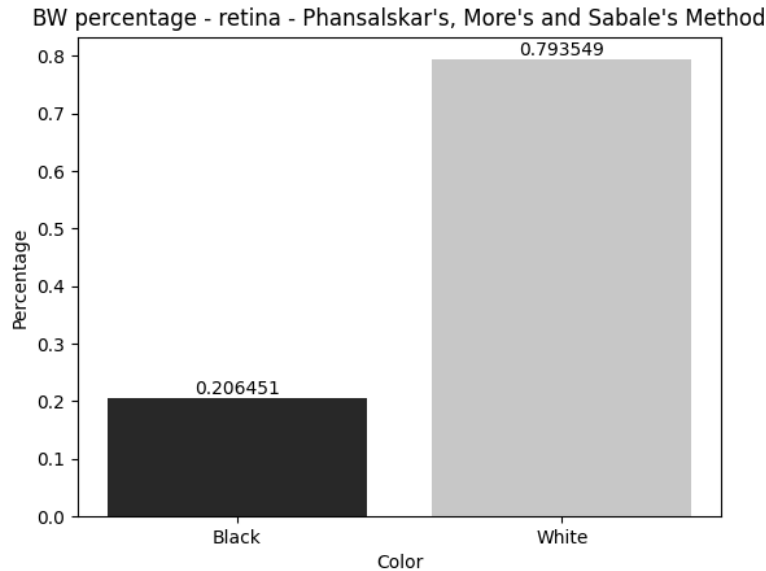
Vemos que é bem fácil distinguir formas na imagem, e claro, vai funcionar muito bem com textos e formas simples como no fiducial e no sonnet, já que objeto e fundo são facilmente definidos por identificações “mais humanas” (as formas da imagem).

7 Método de Phansalskar, More e Sabale

Sendo um método refinado do método de Sauvola e Pietaksinen, ele apresentará resultados similares, com mais opções de refinamento permitindo que sejam atingidos resultados espetaculares para a binarização de imagens, com pouquíssima perda de detalhes e grande precisão na identificação de mudança de contextos.



Método de Phansalskar, More e Sabale
($n = 25$; $k = -0, 12$; $R = 0, 6$; $p = 1$; $q = 100$)



Porcentagem da coloração da imagem binária

Observe que é muito interessante a maneira que esse método consegue separar os traços da retina de maneira muito boa (claro que meus parâmetros não são os melhores), acredito que com um refinamento melhor seja possível eliminar todo o ruído e manter somente os traços da retina. Inclusive esse método foi criado em um contexto de estudar citologia e lendo o artigo [Adaptive Local Thresholding for Detection of Nuclei in Diversely Stained Cytology Images - Neerad Phansalkar, Sumit More, Ashish Sabale, Dr. Madhuri Joshi](#) fiquei realmente surpreso o quão melhor seus resultados foram em relação ao método de Sauvola e Pietaksinen.

8 Métodos de Contraste, Média e Mediana

Como já mencionado, o método de contraste é muito similar ao método de Bernsen e se encaixa em aplicações bem parecidas (e.g. QR codes, imagens OCT).

O método da média acaba sendo bem útil uma vez que ele é um método de suavização de imagens, ele funciona reduzindo a quantidade de variação de intensidade entre pixels próximos e geralmente é utilizado para reduzir ruído de uma imagem.

O método da mediana é um filtro de ruído, assim como o da média, porém geralmente faz um trabalho mais limpo de desejado no quesito de preservar qualidade e detalhes da imagem.

Leitura sobre o método de Bernsen / Contraste: [Implementation of Bernsen's Locally Adaptive Binarization Method for Gray Scale Images - Can Eyupoglu](#)

Leitura sobre o método da média: [HIPR2 - Mean Filter](#)

Leitura sobre o método da mediana: [HIPR2 - Median Filter](#)

9 Histogramas

9.1 Fiducial

Aqui temos uma imagem de baixo contraste (maioria dos pixels menores que 80), apesar de existir uma grande concentração na porção direita do histograma, isso se deve a grande quantidade de detalhes pretos dos códigos na imagem e não a uma boa distribuição de contraste da imagem.

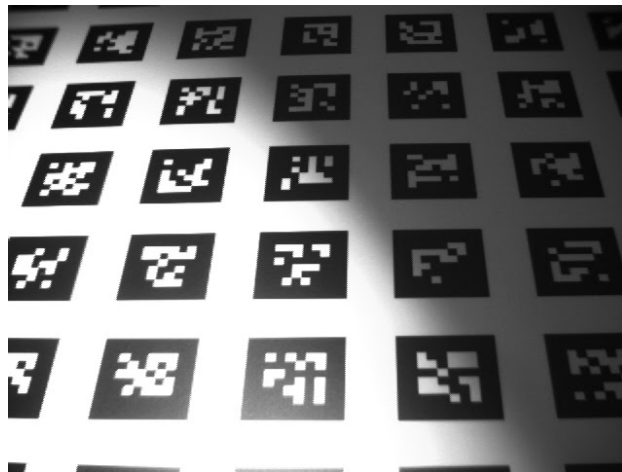
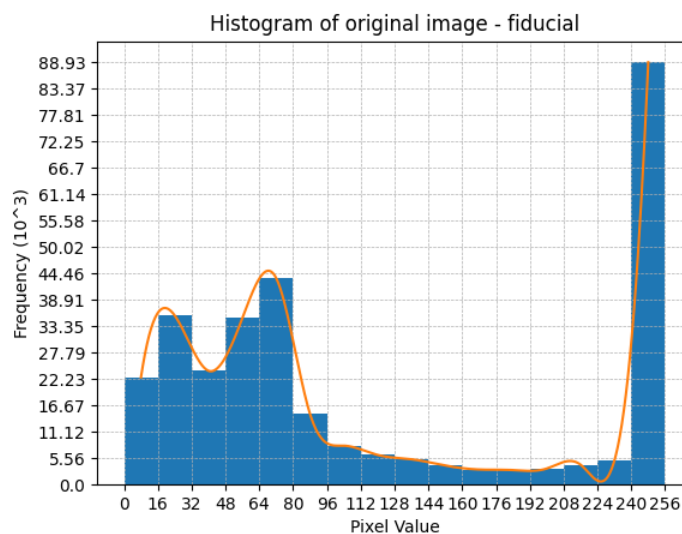


Imagem original



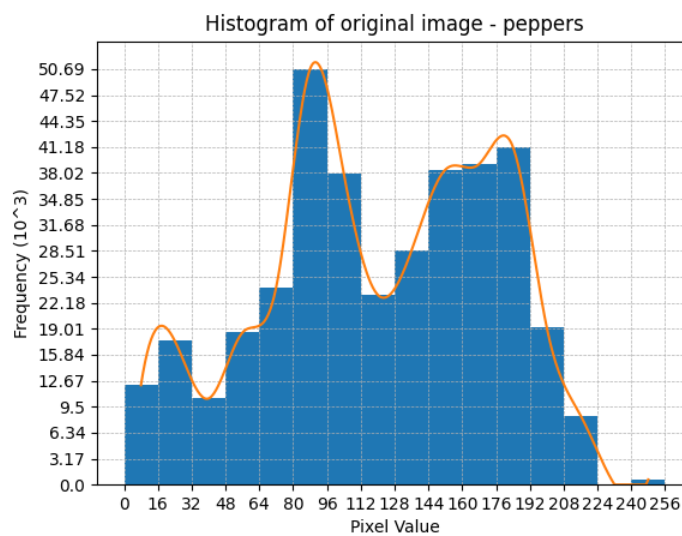
Histograma de Contraste

9.2 Peppers

Aqui temos uma imagem de médio contraste, com uma maior concentração no centro do histograma, tornando uma boa imagem para identificação de sombras e detalhes.



Imagem original



Histograma de Contraste

9.3 Sonnet

Vemos um excelente exemplo de alto contraste nessa imagem, apesar da presença de texto os níveis de cinza estão bem distribuídos.

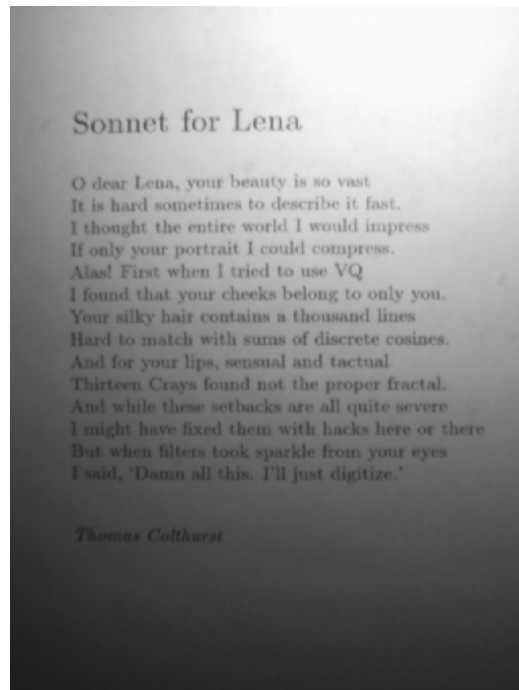
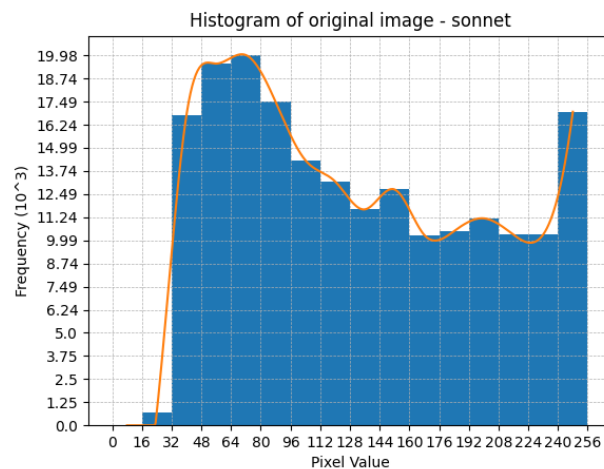


Imagem original



Histograma de Contraste

9.4 Baboon

Outro bom exemplo de alto contraste nessa imagem.

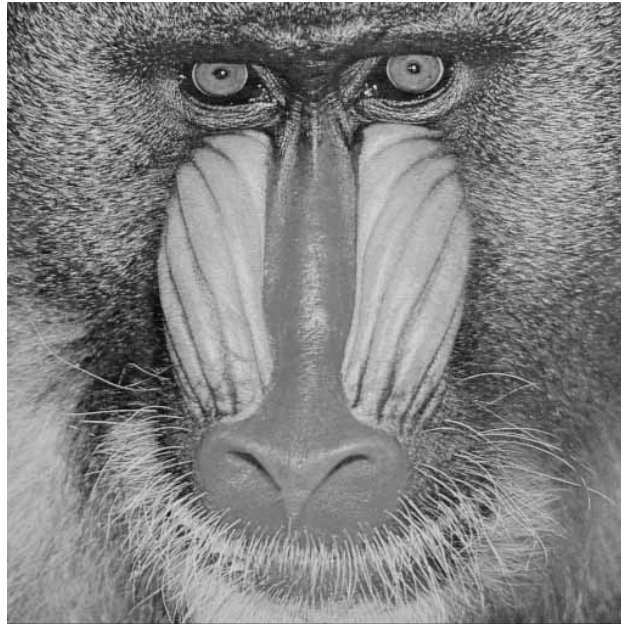
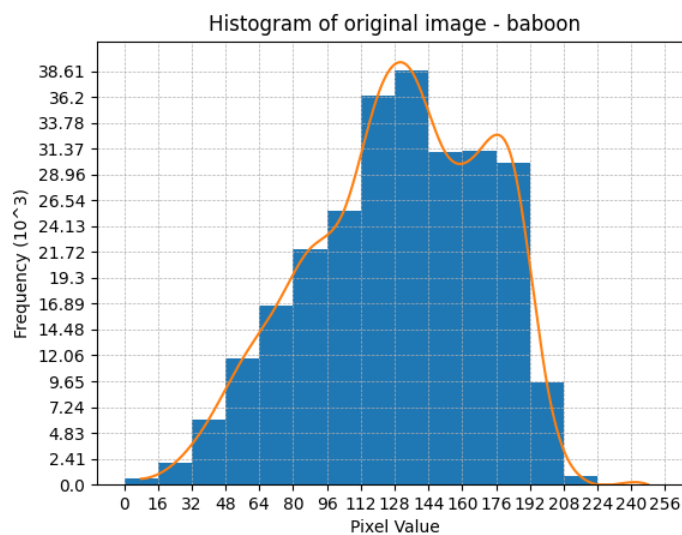


Imagem original

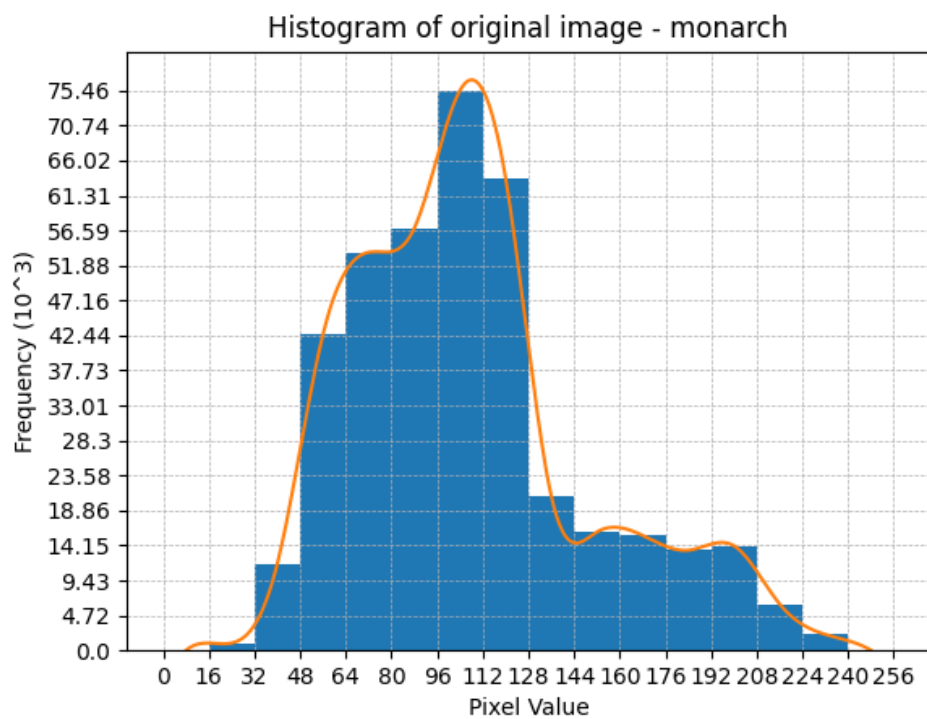


Histograma de Contraste

9.5 Monarch



Imagem original

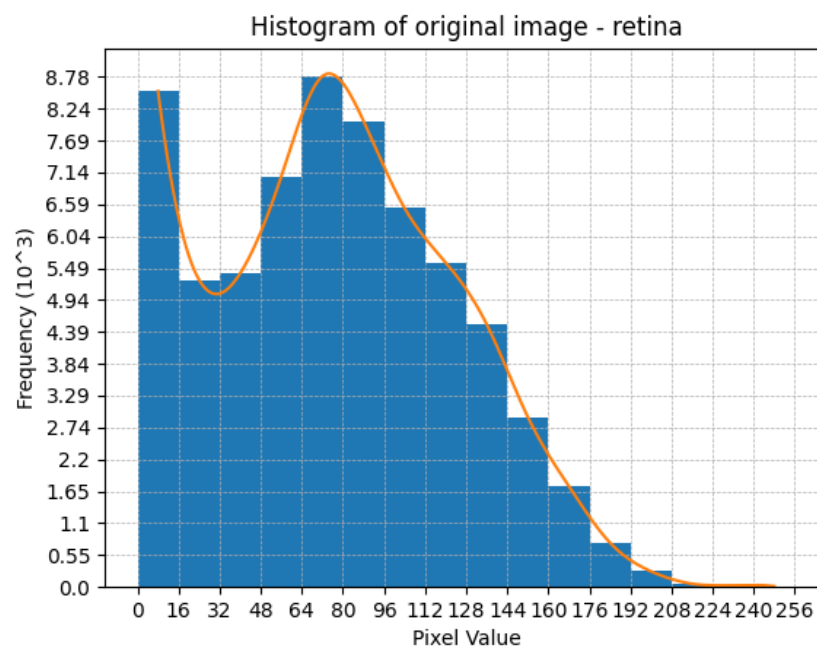


Histograma de Contraste

9.6 Retina



Imagem original



Histograma de Contraste

9.7 Wedge

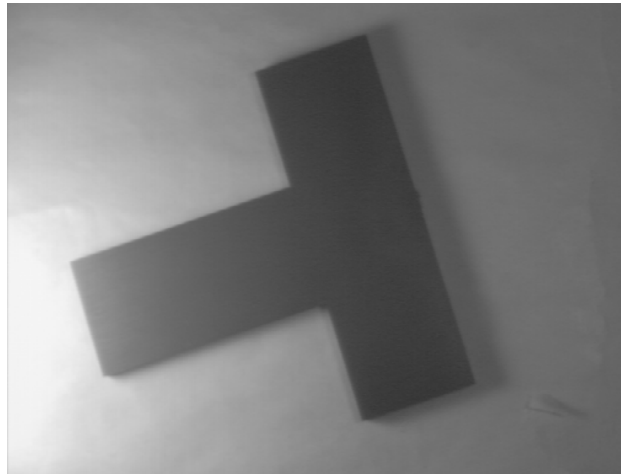
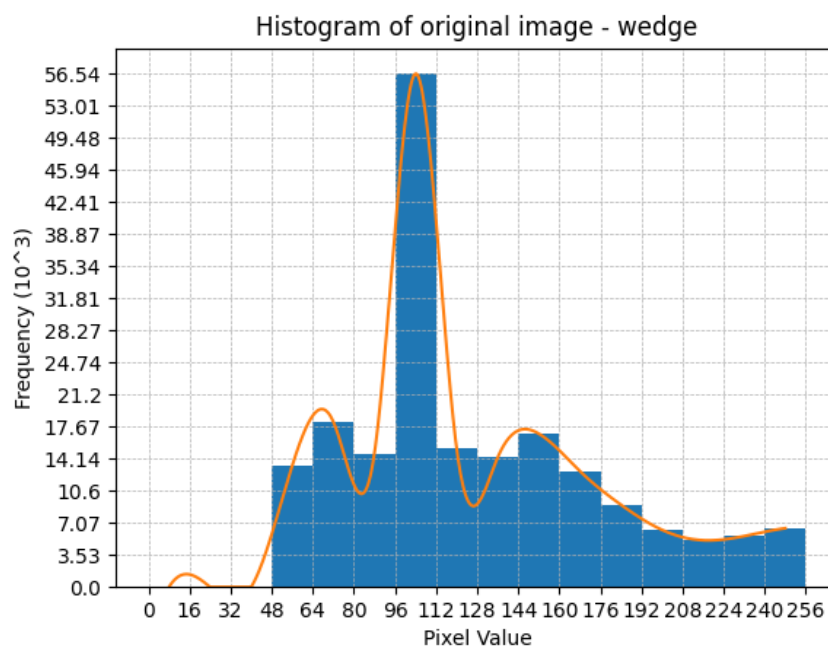


Imagem original



Histograma de Contraste