

Trabalho 3 - Técnicas de Limiarização de Imagens em Escala de Cinza

Leonardo de Sousa Rodrigues – RA: 220063

1 Introdução

Esse trabalho teve como objetivo o estudo, a implementação e a experimentação de técnicas de limiarização de imagens monocromáticas. Cada uma das técnicas utiliza um mecanismo diferente para criar uma imagem binária, na tentativa de separar objetos (*foreground*) do fundo (*background*). As técnicas implementadas podem ser consultadas na tabela 1.

Técnica	Nome para execução
Global	global
Bernsen	bernsen
Niblack	niblack
Sauvola e Pietaksinen	sauvola
Phansalskar, More e Sabale	phansalskar
Contraste	contraste
Média	media
Mediana	mediana

Tabela 1: Técnicas para limiarização de imagens utilizadas no trabalho e os respectivos nomes utilizados como argumentos para execução do programa.

2 Execução

O arquivo `limiar.py`, ao ser executado, recebe 3 argumentos posicionais, conforme a tabela 2.

Argumento	Descrição
IMAGEM	Arquivo de imagem a ser processado.
SAIDA	Nome do arquivo em que a imagem de saída será salva.
METODO	Método de limiarização a ser utilizado, dentre as opções da tabela 1.

Tabela 2: Argumentos do programa `limiar.py` e suas descrições.

Os parâmetros utilizados por cada técnica são determinados pelo arquivo `param.py`.

Um exemplo de execução do programa:

```
python limiar.py entrada/baboon.pgm saida/baboon.png sauvola
```

Nesse caso, a imagem `entrada/baboon.pgm` será limiarizada com a técnica de Sauvola e Pietaksinen e o resultado será salvo em `saida/baboon.png`.

3 Implementação

3.1 Entrada e Saída

A leitura e escrita das imagens em escala de cinza foram feitas através das funções `imread()` e `imwrite()` do OpenCV.

3.2 Técnicas de Limiarização

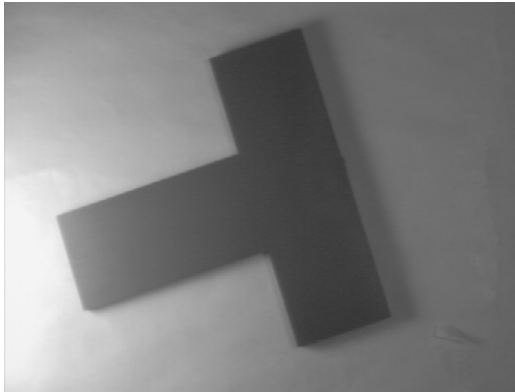
As técnicas de limiarização foram implementadas de maneiras variadas. Para o caso de limiarização global, um simples comando vetorizado do Numpy foi o suficiente. Para a técnica de Niblack e para a

de Sauvola e Pietaksinen, foram utilizadas funções prontas do módulo `ximgproc` do `OpenCV`.

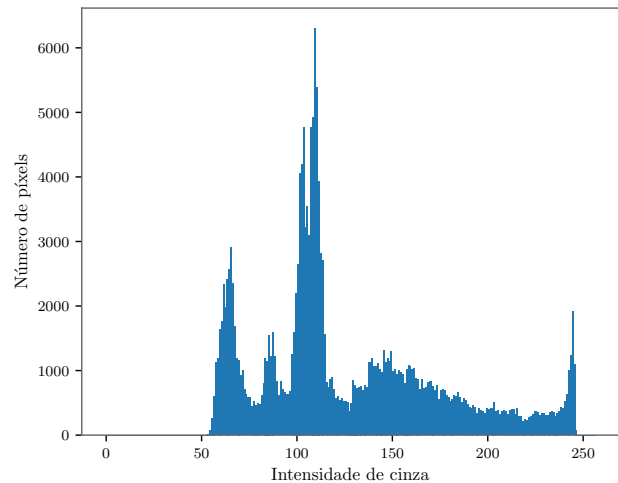
As demais técnicas foram implementadas através da função `generic_filter()` do módulo `ndimage` do `SciPy`. Para cada caso, a função recebe uma vizinhança do píxel em análise e aplica sobre ela a função que a técnica descreve para o cálculo do limiar.

Os parâmetros utilizados pela técnica para a sua execução são armazenados no arquivo `param.py`. Nele, existe um dicionário para cada um dos métodos, com os parâmetros como chaves.

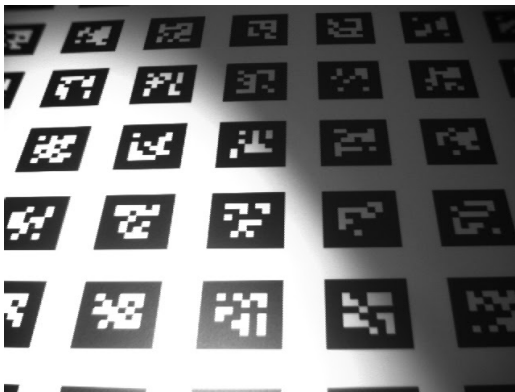
4 Imagens de Entrada



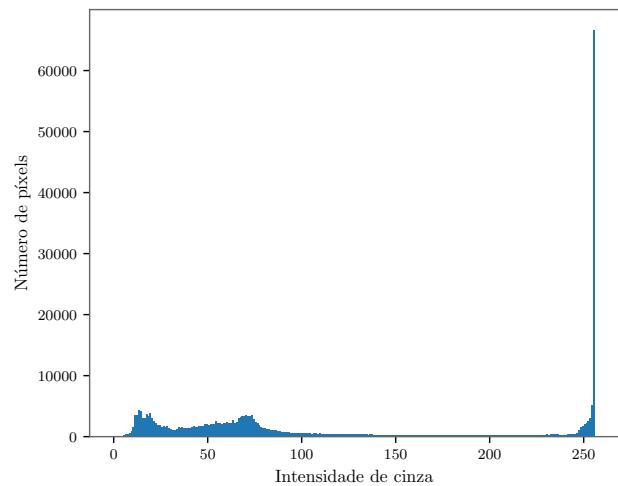
(a) wedge



(b) Histograma de `wedge`.



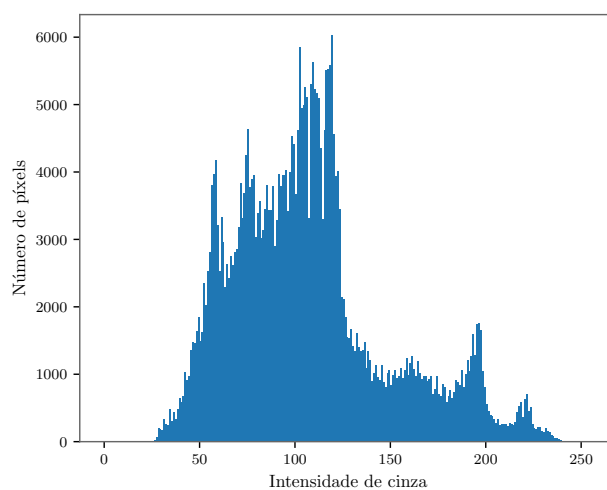
(a) fiducial



(b) Histograma de `fiducial`.



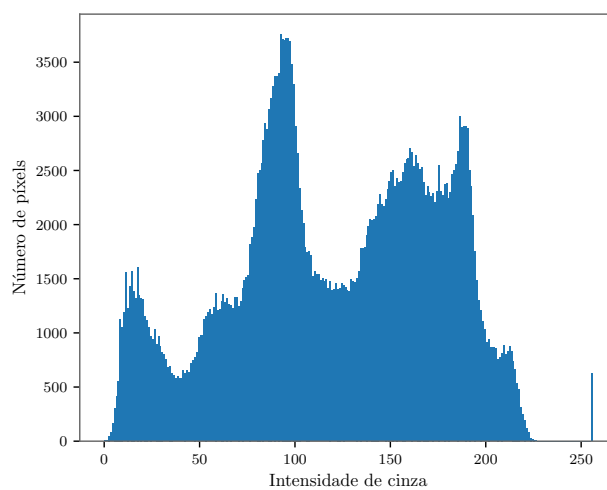
(a) monarch



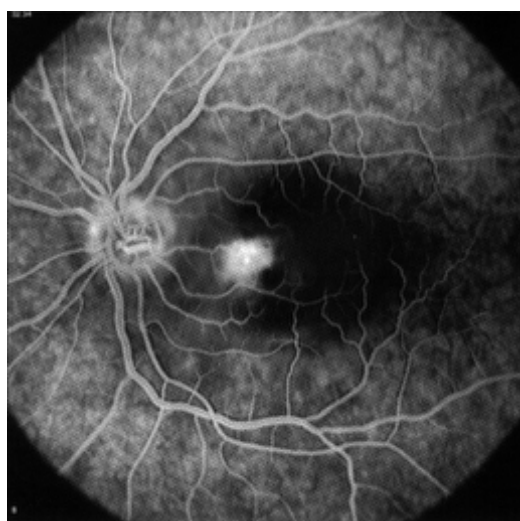
(b) Histograma de monarch.



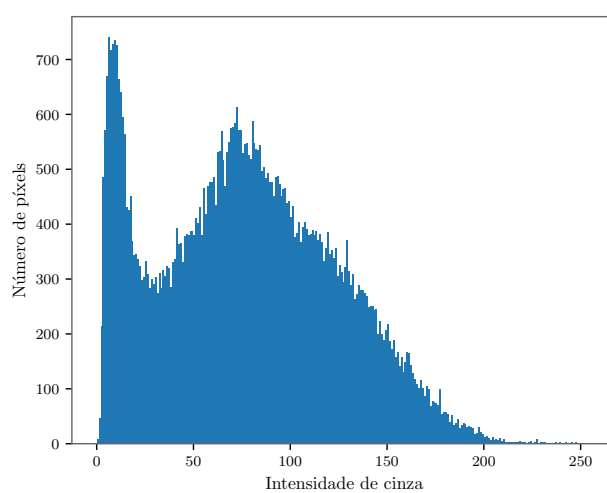
(a) peppers



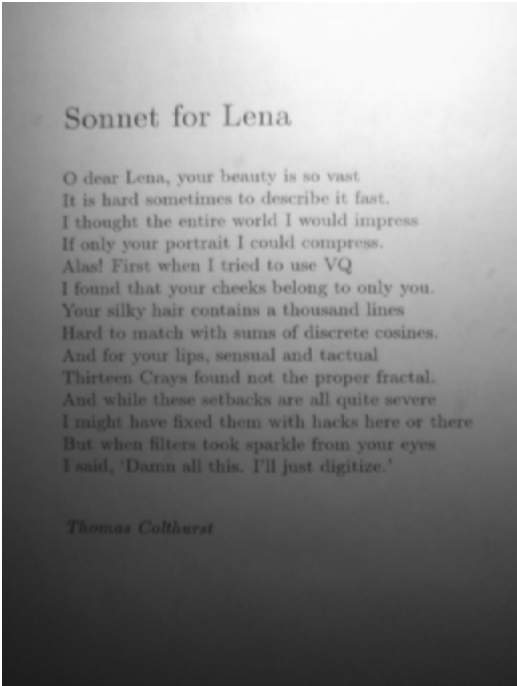
(b) Histograma de peppers.



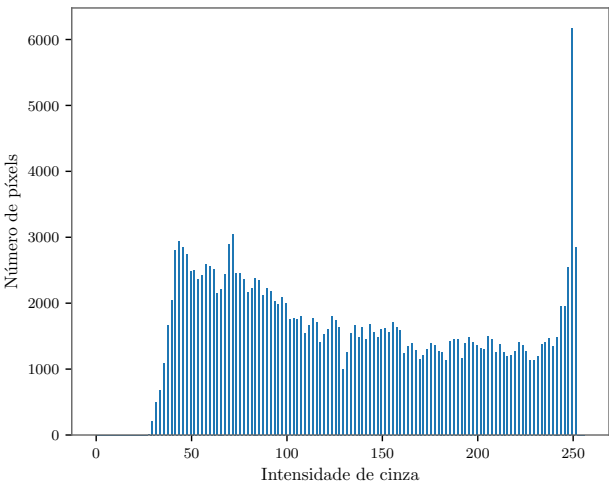
(a) retina



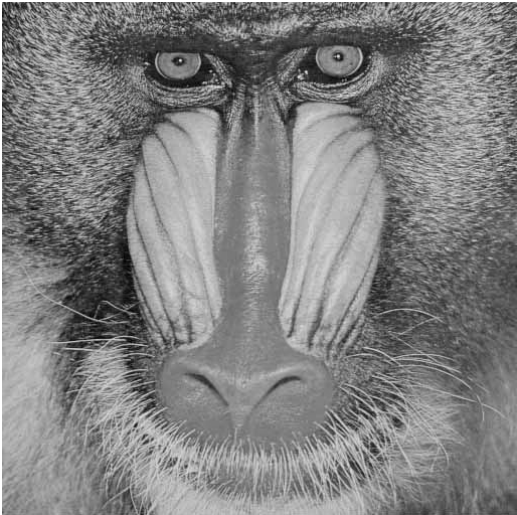
(b) Histograma de retina.



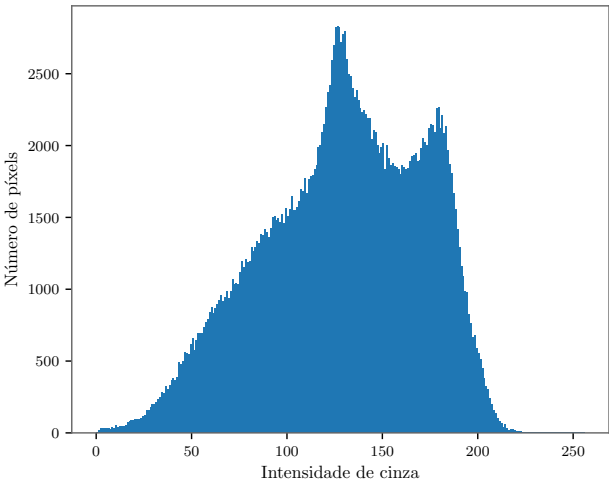
(a) sonnet



(b) Histograma de sonnet.



(a) baboon



(b) Histograma de baboon.

5 Resultados

5.1 Método Global

Os resultados de algumas imagens para o método global pode ser visto na figura 8. Note que esse método é bem simples e, portanto, tem algumas limitações.

No caso da *fiducial*, por exemplo, como a luminosidade da figura não é constante, pontos pretos em uma região podem ser mais claros que pontos brancos em outras. Assim, o resultado é ruim.

Para o caso de uma figura em que o fundo é muito escuro em todos os pontos, no entanto, como é o caso da *peppers*, o método global é eficaz, separando corretamente os objetos da figura, em branco, do *background*, em preto.



(a) baboon para $T = 128$. 47% de pixels pretos.



(b) fiducial para $T = 75$. 50% de pixels pretos.



(c) peppers para $T = 40$. 10% de pixels pretos.

Figura 8: Resultados para o método global.

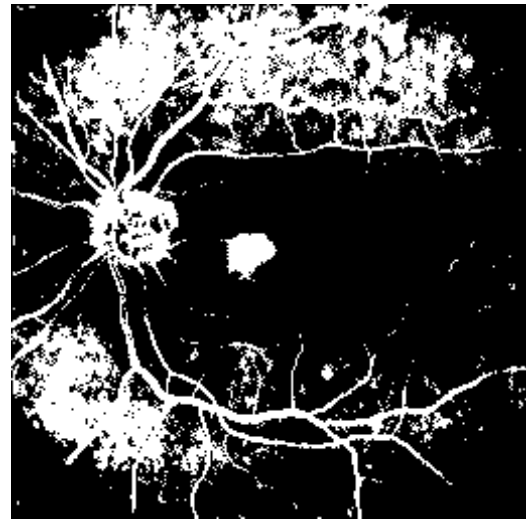
5.2 Método de Bernsen

O limiar no método de Bersen é definido a partir da media entre os valores máximo e mínimo em uma vizinhança. Assim, espera-se que em imagens com conteúdo preto e branco, como a **sonnet**, mesmo com a luminosidade não uniforme, seja possível a limiarização. De fato, na figura 9, nota-se que o método tem um resultado razoável em separar o texto do fundo, mas as bordas da imagem são caóticas.

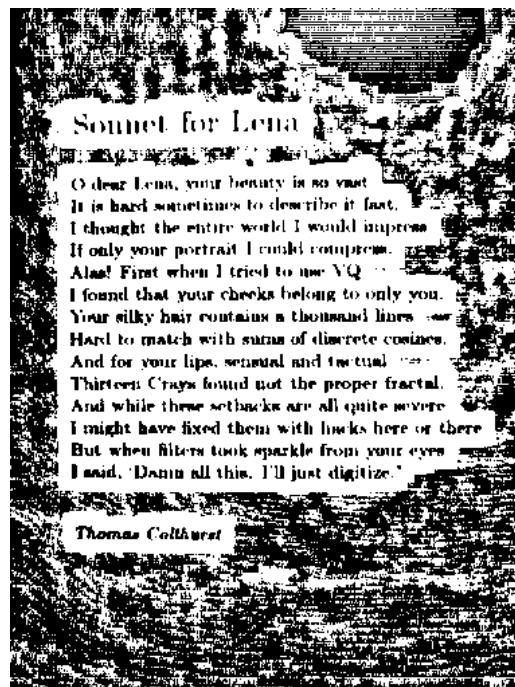
Para imagens mais delicadas, no entanto, como é o caso da **retina**, esse método não apresenta um bom resultado, já que a diferença entre o objeto desejado e o fundo da imagem é mais sutil devido à textura.



(a) peppers para $n = 101$. 47% de pixels pretos.



(b) retina para $n = 81$. 74% de pixels pretos.



(c) sonnet para $n = 21$. 44% de pixels pretos.

Figura 9: Resultados para o método de Bernsen.

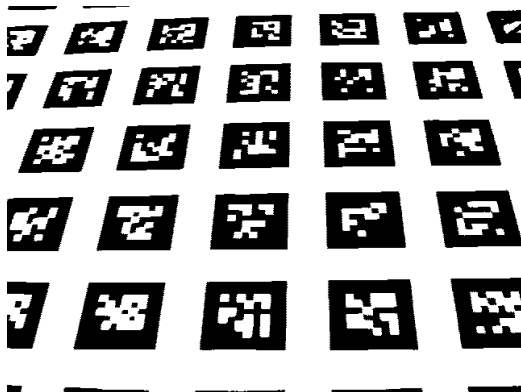
5.3 Método de Niblack

A ideia por trás do método de Niblack leva em consideração a média μ da vizinhança do píxel e o desvio padrão σ . A teoria diz que, assumindo uma distribuição normal, a equação $T = \mu + k \sigma$ seleciona apenas uma faixa de valores da curva normal de acordo com o parâmetro k .

Os resultados podem ser vistos na figura 10. Para cada figura, o k foi variado para encontrar o melhor resultado. Note que o método é perfeitamente capaz de distinguir os símbolos na imagem fiducial. Além disso, faz um trabalho muito melhor que o método de Bernsen em distinguir as ramificações do fundo no caso da retina.

Para o caso da wedge, o resultado é razoável, dado que a luminosidade da imagem é muito desigual. O método não lida bem com as bordas da imagem, que fica caótica e falha em distinguir a região de sombra.

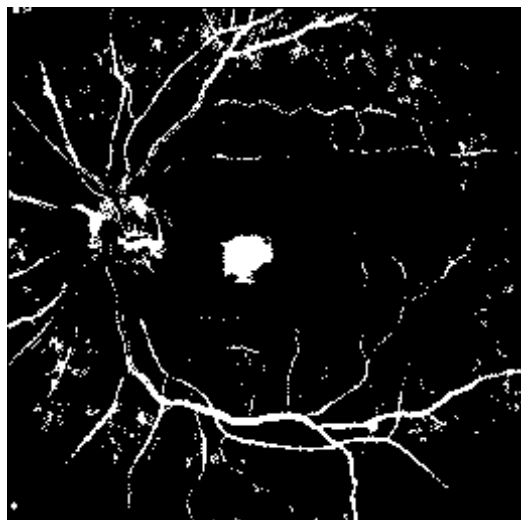
Por último, para a monarch, o método é muito simples e não consegue um resultado satisfatório para nenhum valor de k .



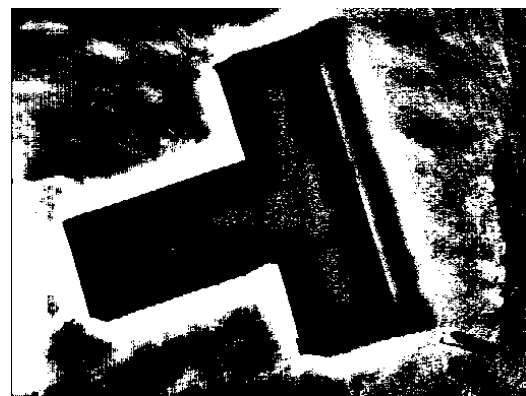
(a) fiducial para $n = 101$ e $k = -0.5$. 34% de píxels pretos.



(b) monarch para $n = 31$ e $k = 0.4$. 73% de píxels pretos.



(c) retina para $n = 71$ e $k = 1.3$. 93% de píxels pretos.



(d) wedge para $n = 71$ e $k = 0.3$. 71% de píxels pretos.

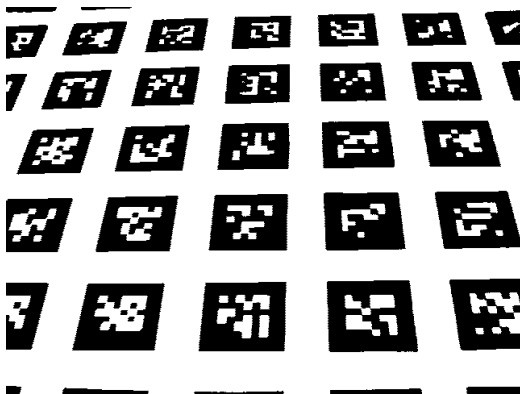
Figura 10: Resultados para o método de Niblack.

5.4 Método de Sauvola e Pietaksinen

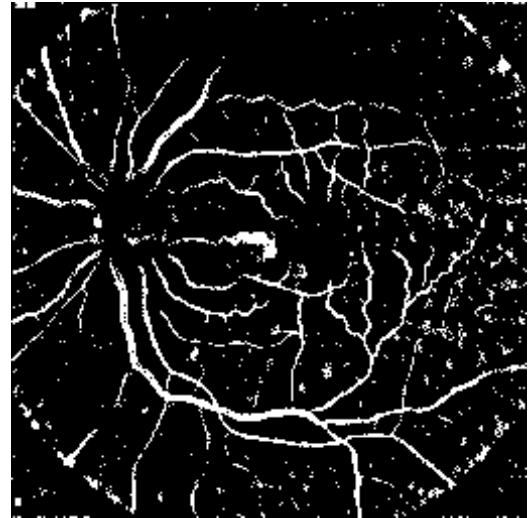
O método de Sauvola e Pietaksinen é mais elaborado que o de Niblack. Com um parâmetro extra, consegue resultados melhores para *retina*, como mostra a figura 11.

Para a *wedge*, o método lida de forma eficaz com o problema de sombra e deixa homogênea a borda da imagem. Embora não consiga diferenciar a parte interna do objeto do fundo, já que os dois apresentam certa homogeneidade, essa técnica define bem as bordas do *foreground*.

Da mesma forma, para a *sonnet*, o método diferencia as letras do fundo e ainda consegue diferenciar as bordas, deixando-as homogêneas.



(a) fiducial para $n = 101$, $k = 0.1$ e $R = 200$. 36% de pixels pretos.



(b) sauvola para $n = 15$, $k = -0.4$ e $R = 128$. 91% de pixels pretos.

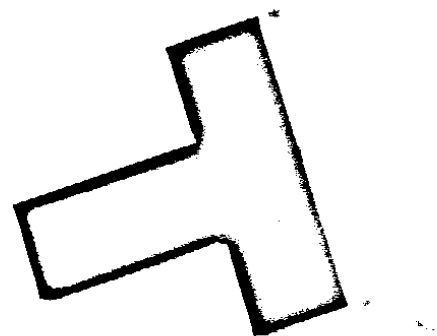
Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crayons found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst



(c) sonnet para $n = 9$, $k = 0.07$ e $R = 200$. 6% de pixels pretos.



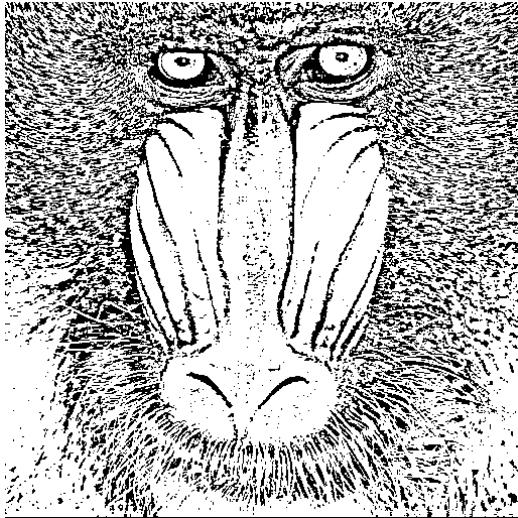
(d) wedge para $n = 31$, $k = 0.1$ e $R = 128$. 5% de pixels pretos.

Figura 11: Resultados para o método de Sauvola e Pietaksinen.

5.5 Método de Phansalskar, More e Sabale

O método de Phansalskar, More e Sabale é o que tem mais parâmetros entre os analisados nesse trabalho. O grande número de opções que isso proporciona faz com que ele consiga lidar melhor com imagens em que a diferença entre o *foreground* e o *background* é sutil. No caso da *monarch*, a técnica faz um excelente trabalho, como pode ser visto na figura 12.

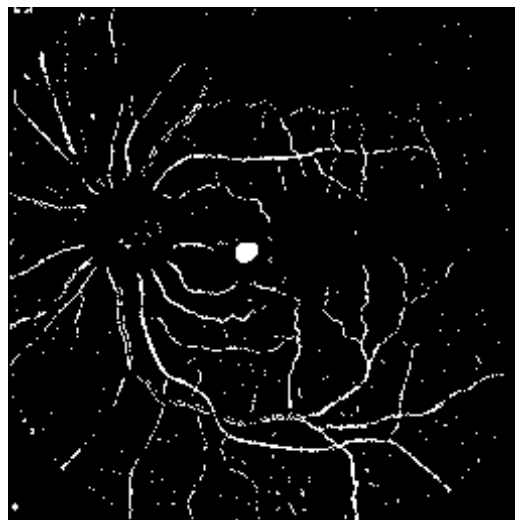
Além disso, o método faz um trabalho ainda melhor que o de Sauvola e Pietaksinen para a *retina*. Apesar de alguns ruídos pontuais, as ramificações da imagem são corretamente detectadas e destacadas do resto da figura.



(a) baboon para $n = 15$, $k = 0.1$, $R = 128$, $p = 4$ e $q = 0.1$. 29% de pixels pretos.



(b) monarch para $n = 31$, $k = 0.9$, $R = 128$, $p = 4$ e $q = 0.1$. 6% de pixels pretos.



(c) retina para $n = 7$, $k = -0.3$, $R = 1024$, $p = 2$ e $q = 0.05$. 96% de pixels pretos.

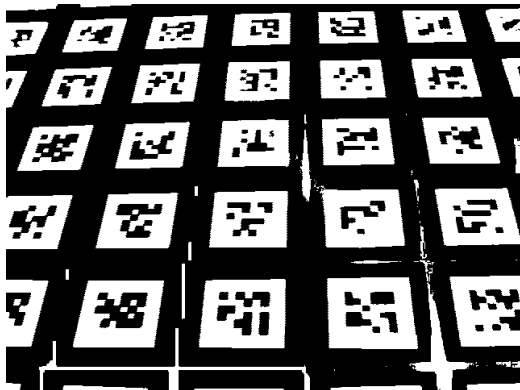
Figura 12: Resultados para o método de Phansalskar, More e Sabale.

5.6 Método do Contraste

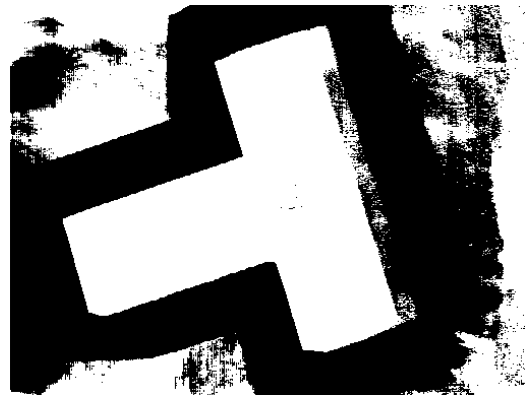
O método do contraste é o negativo do método de Bernsen, já que, se o píxel estiver acima da média entre o máximo e o mínimo da vizinhança, ele é considerado como *background* e, caso contrário, *foreground*.

Os resultados podem ser vistos na figura 13. Note no caso da *fiducial* que com um valor específico de tamanho de vizinhança, é possível identificar corretamente os símbolos, mas há muito ruído entre eles.

A *wedge*, por sua vez, teve suas bordas identificadas, mas com muitos problemas na área de sombra e bordas caóticas.



(a) *fiducial* para $n = 37$. 63% de píxels pretos.



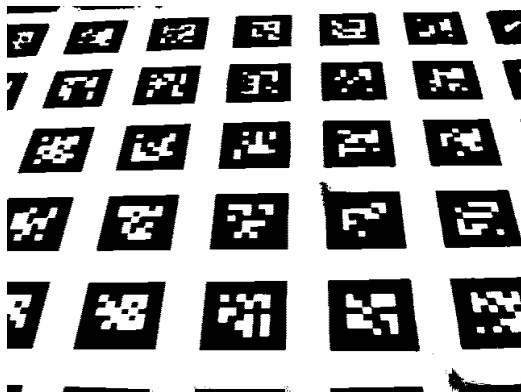
(b) *wedge* para $n = 91$. 52% de píxels pretos.

Figura 13: Resultados para o método do contraste.

5.7 Método da Média

Para o método da média, espera-se imagens com conteúdo preto e branco sejam bem limiarizadas, assim como para o método de Bernsen. Isso porque a análise é local e a luminosidade heterogênea não causa grandes prejuízos. Note da figura 14 que a *fiducial* e a *sonnet* têm resultados razoáveis nas regiões de interesse, apesar da simplicidade do método.

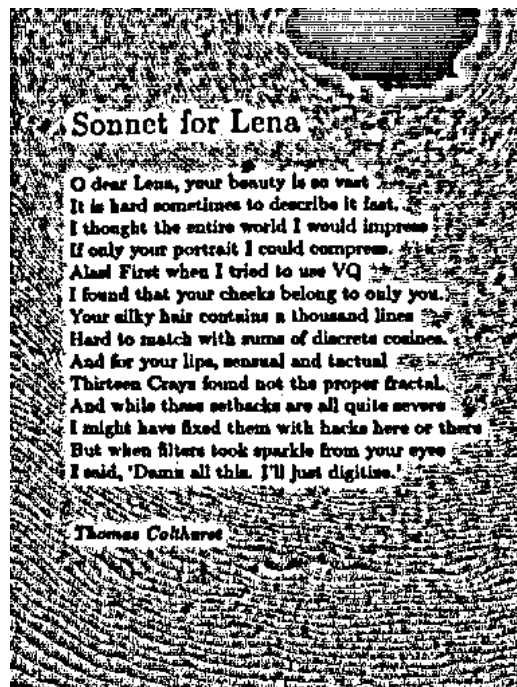
No caso da *peppers*, por outro lado, o resultado é ruim, já que a técnica produz um resultado com alta fração de pixels pretos e o esperado para essa imagem é o oposto.



(a) *fiducial* para $n = 91$. 36% de pixels pretos.



(b) *peppers* para $n = 121$. 48% de pixels pretos.



(c) *sonnet* para $n = 13$. 43% de pixels pretos.

Figura 14: Resultados para o método da média.

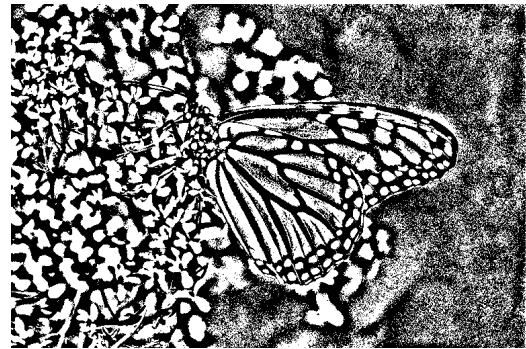
5.8 Método da Mediana

O método da mediana não produz resultados de qualidade mesmo com a variação do tamanho da vizinhança. Note da figura 15 que a *sonnet* é extremamente ruidosa tanto na região de interesse como nas bordas.

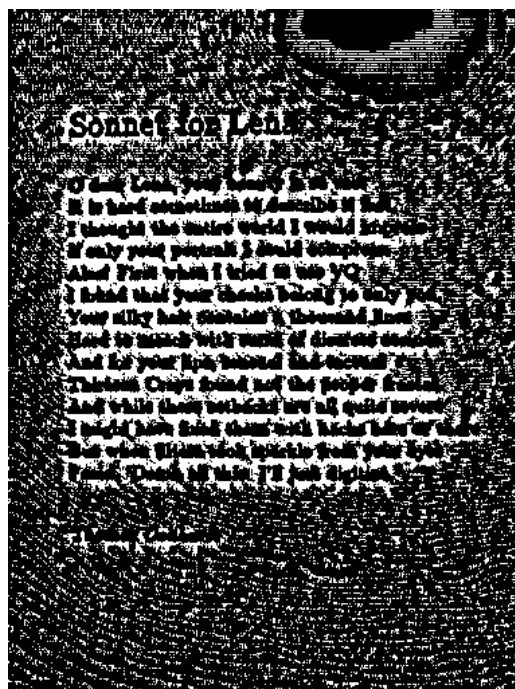
Para a baboon e monarch, não é possível utilizar essa técnica para distinguir o objeto do *background*.



(a) baboon para $n = 41$. 51% de pixels pretos.



(b) monarch para $n = 31$. 56% de pixels pretos.



(c) sonnet para $n = 15$. 74% de pixels pretos.

Figura 15: Resultados para o método da mediana.

6 Conclusão

Durante o desenvolvimento do trabalho, foi possível aprofundar os conhecimentos nas diferentes técnicas de limiarização e identificar quais delas são mais adequadas para algumas situações.

A limitação da técnica global de limiarização pôde ser vista na prática, ainda que seja uma opção interessante para figuras uniformes com fundo preto. Além disso, a complexidade do método de Phansalskar, More e Sabale, devido ao alto número de parâmetros, mostrou resultados de fato melhores que as outras técnicas, conseguindo limiarizar com qualidade a figura **monarch**.

Por fim, é válido ressaltar que, para cada figura, a combinação ótima de parâmetros é diferente, de forma que é necessária a experimentação com um caso específico para obter bons resultados.