

Processamento de Imagens em um Caso de *Optical Character Recognition*

1. Introdução

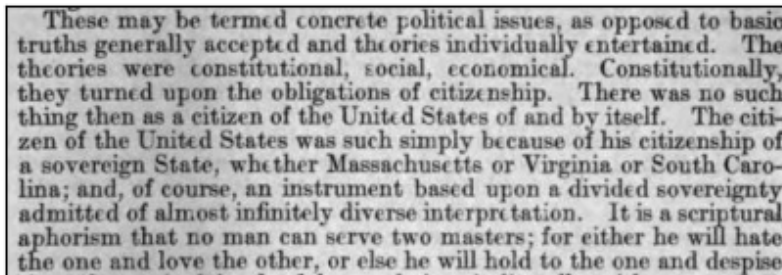
O processamento de imagens desempenha um papel fundamental em diversas áreas, por exemplo: Reconhecimento de Padrões e Visão Computacional. Duas técnicas importantes nesse contexto são o Processamento Morfológico e o uso de Momentos Invariantes. Este relatório explora essas abordagens e suas aplicações. Este relatório trata de uma atividade consistente na aplicação de ambas para realizar uma tarefa de Reconhecimento Óptico de Caracteres.

O Reconhecimento Óptico de Caracteres (*Optical Character Recognition*, OCR) é uma tecnologia fundamental que visa a identificação automática de texto em imagens ou documentos digitalizados. O processamento morfológico desempenha um papel significativo no pré-processamento de imagens para melhorar a precisão e a robustez dos sistemas OCR. Este relatório examina a aplicação do processamento morfológico em casos específicos de OCR.

2. Metodologia

A tarefa a ser realizada nesta atividade consiste em explorar técnicas de processamento de imagem para investigar melhorias nos resultados do OCR. Par isso, foram utilizadas as imagens da Figura 1.

ex-ocr-1.JPG (180, 515)



ex-ocr-2.JPG (284, 591)

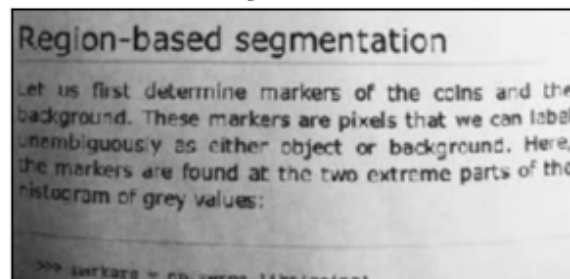


Figure 1: Imagens utilizadas como insumo da tarefa

3. Processamento de Imagem

O processamento morfológico é uma técnica que analisa e transforma a forma e a estrutura das imagens. Pode ser utilizado para operações de pré-processamento, segmentação e reconhecimento de padrões. Dentre elas, as mais comuns incluem Erosão, Dilatação, Abertura e Fechamento:

- Erosão: Remove *pixels* da borda dos objetos, útil na remoção de ruídos e separação de objetos próximos.
- Dilatação: Adiciona *pixels* à borda dos objetos, útil na união de objetos próximos e preenchimento de pequenos espaços.
- Abertura: Combina erosão seguida por dilatação, útil na remoção de ruídos e na separação de objetos conectados.
- Fechamento: Combina dilatação seguida por erosão, útil na união de objetos próximos e no preenchimento de pequenos espaços.

A extração do texto a partir da imagem foi realizada pela ferramenta Tesseract. Ademais, para realizar esta tarefa, foi utilizada a linguagem Python em conjunto com as seguintes bibliotecas: Numpy, PIL, OpenCV, Matplotlib e Pytesseract.

4. Resultados

De posse das imagens da Figura 1, realizamos o OCR e obtivemos os seguintes resultados, respectivamente:

- ' These may be termed concrete political issues, as opposed to basic truths generally accepted and theories individually entertained. Theories were 'constitutional, racial, economical. Constitutionally, they turned upon the obligations of citizenship. 'There was no such thing then as a citizen of the United States of and by itself. The citizen of the United States was such simply because of his citizenship of a sovereign State, whether Massachusetts or Virginia or South Carolina; and, of course, an instrument based upon a divided sovereignty admitted of almost infinitely diverse interpretation. It is a scriptural aphorism that no man can serve two masters; for either he will hate the one and love the other, or else he will hold to the one and despise the other.''
- ' nine markers of the contours and the markers are pixels that we can label either object or background. Here, at the two extreme parts of the'

Percebemos que o resultado obtido é satisfatório para a primeira imagem mas não é suficientemente satisfatório para a segunda. Assim, coube investigar técnicas aprendidas em sala de aula ao longo do semestre letivo buscando melhorar esse resultado. Dentre as técnicas investigadas, foram selecionadas as seguintes, cujos resultados estão expostos na Figura 2:

1. *Gaussian Blur*
2. *Median Blur*
3. *Erosão*
4. *Dilatação*
5. *Treshhold*
6. *Adaptative Treshold*

Também experimentamos combinações dos métodos supracitados (e.g.: Aplicar dilatação ao resultado da erosão). De posse dos resultados, foi aplicada a solução de OCR para realizar a extração de texto, seguido de uma análise comparativa para determinar qual técnica, ou conjunto de técnica, resultou no maior aumento de qualidade.

Dada a análise comparativa, o melhor resultado foi obtido quando utilizada apenas a técnica de *Gaussian Blur*, exposto a seguir de maneira identada:

based segmentation
ine markers of the coins and the
markers are pixels that we can label
either object or background. Here,

d at the two extreme parts of the

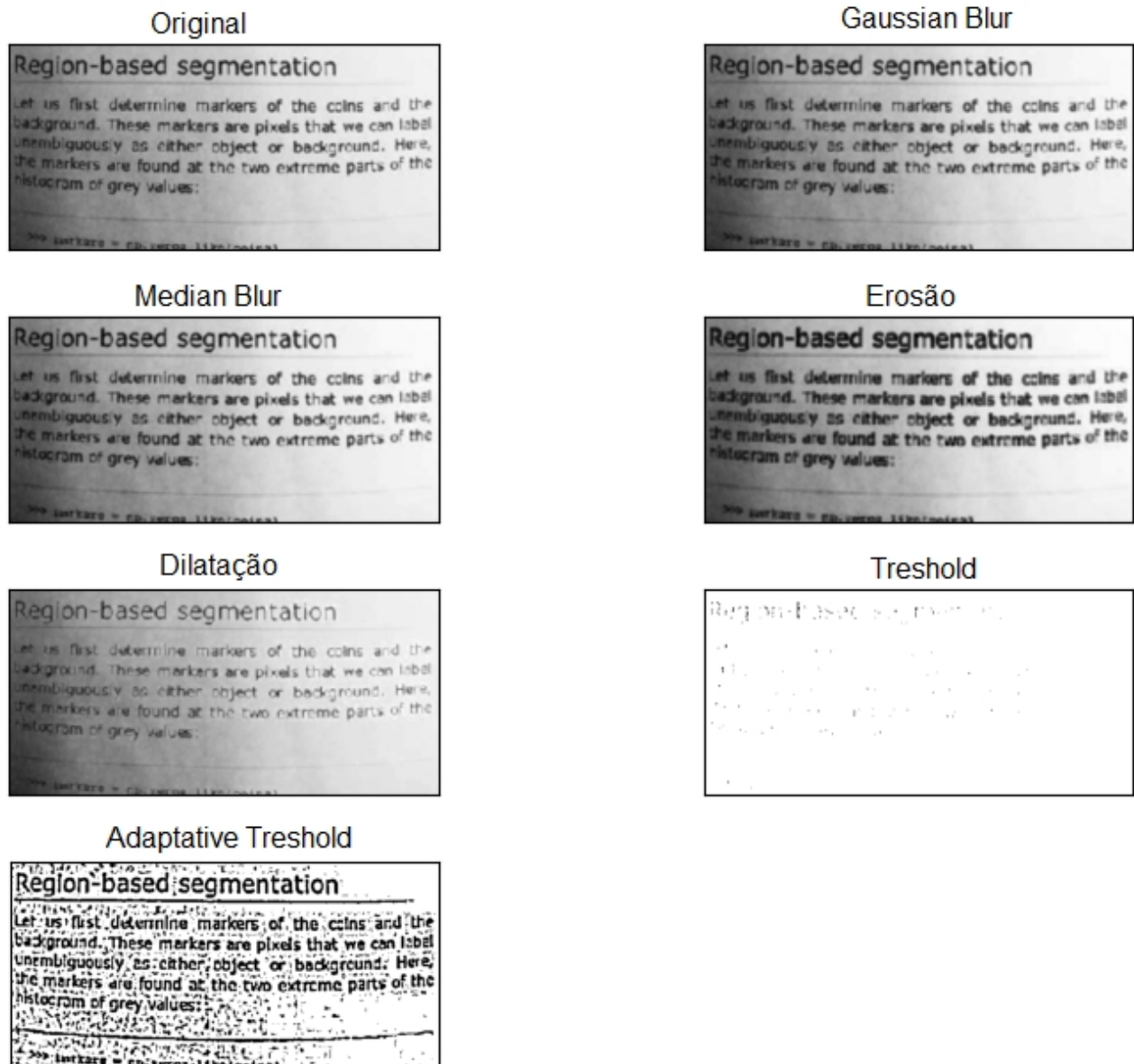


Figure 2: Aplicação de técnicas de processamento de imagem

5. Conclusões

O processamento morfológico desempenha um papel crucial no aprimoramento do desempenho de sistemas OCR, oferecendo ferramentas valiosas para pré-processamento de imagens. A aplicação de operações morfológicas pode melhorar a segmentação de caracteres, remover ruídos e garantir a precisão na interpretação de texto em imagens. A escolha cuidadosa dos parâmetros e uma compreensão profunda dos desafios específicos em aplicações OCR são essenciais para otimizar a eficácia do processamento morfológico.

Anexo

Detalhes da implementação utilizando código em Python para processamento das imagens:

```
import cv2

import numpy as np
import seaborn as sns

from math import sqrt
from random import choice

from PIL import Image
from matplotlib import pyplot as plt

%matplotlib inline

inverte = lambda i: cv2.bitwise_not(i)

gblur = lambda i: cv2.GaussianBlur(i, (3,3), 0)
mblur = lambda i: cv2.medianBlur(i, 3, 0)

kernel = np.ones((3,3), np.uint8)
erosao = lambda x: cv2.erode(x, kernel, iterations=1)
dilatacao = lambda x: cv2.dilate(x, kernel, iterations=1)

tresh = lambda i: cv2.threshold(i, 50, 255, cv2.THRESH_BINARY)[-1]
otsu = lambda i: cv2.threshold(cv2.GaussianBlur(i,
(5,5),0),0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)[-1]
adtresh = lambda i: cv2.adaptiveThreshold(i,255,cv2.ADAPTIVE_THRESH_MEAN_C,
cv2.THRESH_BINARY,11,2)

def dithering(pixels):
    return Image.fromarray(pixels).convert("P", dither=Image.Dither.FLOYDSTEINBERG)
```