

UNIVERSIDADE FEDERAL DE SERGIPE DEPARTAMENTO DE COMPUTAÇÃO

PROCESSAMENTO DE IMAGENS Beatriz Trinchão Andrade

PROJETO PRÉ OCR

LETÍCIA CENA DOS SANTOS ANY CAROLINY SOUZA SILVA

> SÃO CRISTÓVÃO 2024

Sumário

1. Introdução	3
2. Técnicas Aplicadas	3
2.1. Filtro da Mediana:	3
2.2. Operações Morfológicas (Dilatação e Erosão)	3
3. Código desenvolvido	4
3.2. Análise e Conversão da Imagem	4
3.3. Filtro de Mediana para Remoção de Ruído	4
3.4. Operações de Dilatação e Erosão	5
3.5. Contador de linhas	7
4. Tratamento de erros	9

1. Introdução

Neste trabalho, desenvolvemos um sistema que aplica técnicas de operações morfológicas como erosão e dilatação além do filtro da mediana com o intuito de promover reconhecimento ótico de palavras, colunas, linhas (OCR) utilizando Python 3.6. Isto é feito após a remoção de ruídos na imagem fornecida de maneira a aprimorar a precisão e a legibilidade do processo de reconhecimento.

2. Técnicas Aplicadas

2.1. Filtro da Mediana:

A técnica de filtro de mediana foi aplicada para remover ruídos permitindo uma melhor visualização das letras e obter homogeneidade do fundo branco. Este filtro é particularmente eficaz na remoção de ruídos impulsivos, preservando os contornos das letras e minimizando a perda de detalhes importantes. O filtro da mediana é calculado a partir da aplicação de uma máscara (também conhecida como kernel), de dimensão m e n (largura e altura) em formato de matriz de forma que o resultado do somatório do produto da vizinhança junto ao elemento central da matriz dispostos em cálculo de mediana matemática obtenha-se a predominância dos itens que mais aparecem, no caso das imagens aqui estudadas, o fundo branco.

2.2. Operações Morfológicas (Dilatação e Erosão):

As operações morfológicas desempenharam um papel essencial no processamento das imagens binárias. A dilatação foi empregada para expandir as regiões de interesse, expandindo pixels pretos próximos e preenchendo lacunas, enquanto a erosão foi utilizada para reduzir o tamanho dos objetos, eliminando pequenos ruídos e refinando as formas dos caracteres. Essas técnicas contribuíram para a melhoria da segmentação e da clareza dos elementos textuais na imagem.

3. Código desenvolvido

No código criado algumas operações são realizadas conforme descrição abaixo 3.1. Leitura de Imagem em Formato PBM:

 Implementamos uma função leitor_PBM para ler arquivos de imagem no formato PBM (Portable Bitmap), que é uma representação de imagem binária.

```
recebe o caminho de um arquivo .pbm como entrada e lê todo o seu conteúdo,
retornando uma string contendo as linhas do arquivo. """

def leitor_PBM(arquivo):
    # Abre o arquivo em modo de leitura
    with open(arquivo,'r') as arquivo:
        linhas = arquivo.read()
    # Retorna o conteúdo do arquivo em formato string
    return linhas
```

3.2. Análise e Conversão da Imagem:

 A função parse_string_array analisa a string do arquivo PBM, extrai largura, altura e os pixels da imagem, convertendo-os em um vetor bidimensional.

```
"" parse_string_vetor
recebe uma string contendo o conteúdo de um arquivo .pbm e o analisa para
extrair
largura, altura e os pixels da imagem """

def parse_string_vetor(pbm_string):
    # Divide a string em uma lista de linhas, separadas pelo caractere de quebra
de linha
    linhas = pbm_string.split('\n')

# Remove linhas de comentário que começam com '#' da lista
    linhas = [linha for linha in linhas if not linha.startswith('#')]

# Obtem a largura e altura da imagem a partir da segunda linha e converte
para inteiros

largura, altura = map(int, linhas[1].split())

# Obtem os pixels da imagem a partir da terceira linha
pixels = linhas[2:]

vetor_inteiros = []
for linha in pixels:
    # Converte cada caractere da linha para inteiro e adiciona a lista de
pixels
    vetor_inteiros.extend([int(char) for char in linha])

# Converte a lista de pixels em uma matriz bidimensional
```

```
vetor_convertido = [vetor_inteiros[i:i+largura] for i in range(0,
len(vetor_inteiros), largura)]

return largura, altura, vetor_convertido
```

3.3. Filtro de Mediana para Remoção de Ruído:

• Utilizamos o filtro de mediana para remover ruídos da imagem. A função filtro_mediana percorre a imagem aplicando a mediana na vizinhança de pixels, substituindo o pixel central pelo valor mediano.

```
""" mediana
    calcula a mediana de uma lista de inteiros """

def mediana(lista):
    return int(statistics.median(lista))

""" filtro_mediana
aplica o filtro da mediana a matriz de pixels """

def filtro_mediana(largura, altura, matriz):
    matriz_vizinhos = matriz_auxiliar(matriz)

# Percorre as linhas da matriz (exceto as bordas)
for i in range(l,altura-1):
    # Percorre as colunas da matriz (exceto as bordas)
for j in range(l,largura-1):
    # Obtem os valores dos vizinhos do pixel atual
    vizinhanca = obter_vizinhanca(i,j, matriz_vizinhos)
    novo_pixel = mediana(vizinhanca)
    # Substitui o valor do pixel atual pelo valor da mediana
    matriz[i][j] = novo_pixel

return matriz
```

3.4. Operações de Dilatação e Erosão:

 Implementamos funções para dilatação e erosão de imagens binárias, utilizando o ponto mínimo (erosão) e o ponto máximo (dilatação) da vizinhança para ser utilizado de forma estruturante na nova imagem a ser gerada. Erosão:

```
def erodir_imagem(imagem, elemento):
    # Obtendo as dimensões da imagem e do elemento estruturante
    altura, largura = imagem.shape
    altura_elem, largura_elem = elemento.shape
```

```
# Inicializando uma matriz para a imagem erodida com zeros
imagem_erodida = np.zeros((altura, largura), dtype=np.uint8)

# Calculando o tamanho do padding necessário ao redor da imagem
pad_altura = altura_elem // 2
pad_largura = largura_elem // 2

# Iterando sobre cada pixel da imagem
for i in range(pad_altura, altura - pad_altura):
    for j in range(pad_largura, largura - pad_largura):
    # Calculando o mínimo dos valores resultantes da
multiplicação entre a região da imagem e o elemento estruturante
    if np.min(imagem[i-pad_altura:i+pad_altura+1,
j-pad_largura:j+pad_largura+1] * elemento) == 1:
    # Se o mínimo for 1, define o pixel na imagem erodida
como 1 (branco)
    imagem_erodida[i, j] = 1

return imagem_erodida
```

Dilatação:

Com isto, ao aplicarmos erosão na imagem em questão temos um resultado de diminuição da intensidade de cores e no caso da dilatação temos um aumento de intensidade e área ocupada pelos pixels expandidos.

Erosão

Dilatação

Lorem ipsum dolor si amet, consecteux adipisking elit, sed do eliusmod tempo incididunt ul lapore et dolore magna aliqua Ut enim ad minim veniam, quis nostrut exercitation ullamco laboris nisi ut aliquip er ac commodo consequat. Duis aute irun dolor in reprehendent in volluptate velti esse ullum dolore eu liquat nulla paratur. Excepeur sini occaecat cupidatai non proidents sunt in culpa qui officia deserun

Sed ut personues a voluptatem ascentant a voluptatem ascentant voluptatem ascentant aperam eaque pas quae ab lio invento aperam eaque pas quae ab lio invento ventais et quas architecto beatae vita dicta sunt explicabo. Nemo enim ipas voluptatem qui avoluptatem supernatur anodi aut fugit sed quia consequentur maguores eso qui ratione voluptatem sequinascent. Neque porro quisquami est qui odiorem ipas mi quia dotor est ame consectetur, adipisci velit, sed quia no immiguam ei as modi tempora incidunt i laborie et dotore magnam aliquam quaer quia nostrum exercitationem ultam corpor composito processituri? Quis autem vie eum ium reprehendent qui in ea voluptatem ultima corpor composituri qui exercitatione di in molestia consequatur. Veli illum qui dotorem eur fugiat quo voluptas nulla paratira qui in qui dotorem eur fugiat quo voluptas nulla paratira pura proprieta proprieta proprieta proprieta della p

Ar verti eles el actuarsans el riusto ano proposento y comparto del propose de corrupti quos dolores el quas molesta exceptun sint occaecat: cupidate no provident similique sunt in culpa qui offici deserunt mollita animi, id est laborum « dolorum fuga. El harum quidem rerur radiis est el expedita distincto. Nam liber tempore cum soluta nobis est eligento pono cumque intili impedit quo minus si omnis voluptas assumenda est, omnis otto repellendus. Temporibus autem quibusdar et aut officiis debits aut rerur necessitatibus saepe eveniet ut « non recusandae. Itaque earum rerum tenetur a sapiente delectus. ut reiciendis voluptatibus maiores a consequatur aut perferendis dolori asperiores repellat.

Sed ut perspoiatis unde omns iste neror sit voluptatem accusan doloremque laudantum totam aperiam eague pps quae ab lio inven ventatas et quasi architecto beatae vi contrate produce and a contrate quae voluptatem sit aperiam eague por qui aconsequium um dolores eos qui aconsequium mi dolores eos qui ratione voluptatem si resciunti. Neque porro quisquam est dolorem ipsum quae dolor sit and concessetami adapsio vella sed quae consequium accompanio del produce del produce

At vero eos et accusamus et iustoc diginasismos ducimus qui bland praesentium voluptatum deleniti at accorrupti quos dolores et quas moles excepturi sint occaecati cupiditate i providenti similique sunti no iupid qui off deserunt mollitia animi, id est laborum foliorum fuga. Et harum quidem retactilis est et expedita disinicito. Nam lib empore cum soluta nobito est elige empore cum soluta nobito est elige

Lecen ignem debr sit aust, connectiva nijivaing elli Sel dispet puns et laux veriletim tament. Haccons authen egyl es it aust susper. Supunitus is mile sit aust too lechu vennenin. Deuer it diem mile, feuralem som wi, confinentum sith, Alipans nitus miletim sithem. Superitus en miletim sithem. Superitus en miletim sithem. Superitus en et una. Placellus ennes uns puns, ubitus alipans delar esture enue. Placellus lendroit lecinis unu punnes wais. Alipans which punses wais. Alipans which punses wais. Alipans which in the sith in the superitus with sithight suspens the lecident inter, at confinentum superitus Microson aliban suspen sit ment lecius enuequet, gonite lecunis augue a trans vienu sipuns. Superitire Mintil fight occur atem vienus, dipinisis malessala unus alipans. Sed confinentum mosa alipans. Sed confinentum mosa alipans.

It et aliquem date. Grimper pubs med milo vibre emedie, liber fregist med milo vibre emedie, liber fregist med en milo vibre emilion. medie ense retirals. Here medieme medie ligida, spet intention diem aliquet pretien. Idla fregist assuper treuper. Pracesont planelen too ar nith militie planelen too ar nith militie mescetcher. Hetger are tridique offe, siter bilantis betos. Pellentesque ut betos menis. Vestitutum arie ignomprints in tenetium ouir betos et olivieres pomme celeliin cener; Frence nor munis vet lanter deplies autur vibre Vesilialem sit austi tida kannoti hano winipot accuman man a cen. Madi fisilian munic ac hann sayifi whicha. Viranes aliquet mitistale lieno quis kidapen. Sepandian kidape pellusirapen elementan. Sa malenfe lectra, quis les malén, lei malenfe lectra, qui le malén, lei malenfe lectra, qui le malén, lei malenfe lectra, qui le malenfe lectra, qui la ficilità augus daplant. Hinn latte en esti vel genite. Hois valle chilend lemper genite. Poin ulte lette vel est impliem accuman.

uniform on quant man. Processo as uniform on Quinque lawest dalor unconsiste parts. In depline, ent al most uniform which their union do del, alt amed suchsique since after velocite. Here facilité, them after ce, ence malle super ent almost ell. Cons at sugar est. Integro most susque il felic tingle compantement at manifera est. Here the la man at manifera est. Here the thirds sugaria. Un sent although the such as a part of the lacts. In the major companies to the manifer ell in magan companies when the consistence of the consistence of the sucher elle manifera. Verification this, a comman a dai oil aust, their maniferation of the latter when prefere the super construction of the sucher elle maniferation.

lor: senjer sanijit ed, vel prime val. Milo compe identum maldes. Vascord accurace, tupis id subdes dichalis, edi ui cummob sejian, up primetegue be com um telu. Incorace of dipum milo. Duce mere concept benç et fesiois des milis en. Huca d'inglis man, d. dictum fight. Superation de combe telles um burm beinis, co mande telles um burm beinis, co

Para aumentar a dilatação foi realizado um aumento de kernel, no âmbito da dimensão da máscara, em largura e altura. Com isto, conseguimos um resultado onde é possível visualizar as palavras delimitadas em formato retangular conforme imagem abaixo à esquerda. Na imagem a direita, com ampliando ainda mais a máscara aplicada foi possível deixar mais clara a relação de linhas sendo visualizadas.

tenem ipmen deber all ment tenem ment in the ment tenem and the ment of here wealth-from homes. However, Chapmadon is suite all ment to be been wearth. However, and the ment of the ment

3.5. Contador de linhas

A função abaixo, "contador_linhas" realiza o cálculo de linhas por colunas na imagem. Esta mesma função também atua no reconhecimento do número de colunas existentes na imagem.

```
if vetor imagem[i][c] == 1 and vetor imagem[i+1][c] == 0:
return (max(array_2_col1), max(array_2_col2))
        if vetor_imagem[i][c] == 1 and vetor_imagem[i+1][c] == 0:
   array_3_col1.append(qtd_linhas)
        if vetor_imagem[i][c] == 1 and vetor_imagem[i+1][c] == 0:
   array_3_col2.append(qtd_linhas)
        if vetor_imagem[i][c] == 1 and vetor_imagem[i+1][c] == 0:
```

4. Tratamento de erros

- Tratamento de Borda na Aplicação do Filtro de Mediana
 - Para evitar a leitura das bordas da imagem ao aplicar o filtro de mediana, ajustamos as iterações para que a vizinhança considerasse somente pixels válidos dentro da imagem. Dessa forma, há uma verificação se a linha está dentro dos limites da matriz

- Eficiência Computacional
 - Implementamos algoritmos otimizados para garantir uma execução eficiente, minimizando o tempo de processamento.

Referências Bibliográficas

Rafael C. Gonzalez, Richard C. Woods - Fundamentals of Computer Graphics (Editora A. K. Peters - 3a edição - 2009)

H. Pedrini, W.R. Schwartz. Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações (Editora Thomson Learning - 2008)