

Trabalho MC920 3: Segmentador de Textos

Esdras R. Carmo - RA 170656

18 de maio de 2019

1 Introdução

Exploraremos nesse trabalho operações morfológicas em imagens branco e preto para segmentar regiões de texto e não texto em uma imagem.

O objetivo do algoritmo desenvolvido será separar áreas de texto das áreas de não texto e segmentar palavras na área identificada como texto. Após isso, o algoritmo será capaz de contar a quantidade de linhas e de palavras identificadas.

2 Especificação do Problema

O algoritmo receberá uma imagem binária de entrada I e será capaz de segmentar as palavras presentes em I . Para isso, dividimos em 3 etapas: pré-processamento da imagem, segmentador de textos e segmentador de palavras.

2.1 Pré-processamento da Imagem

A etapa de pré-processamento consiste em uma sequência de operações morfológicas utilizadas de modo a unir as linhas do texto para que possa ser segmentado posteriormente. Nesta etapa também é importante garantir que regiões sem texto não sejam confundidas com regiões com texto.

Primeiramente é realizado duas operações de fechamento, uma em linhas e outra em colunas e realizado a intersecção dos resultados. Após isso, é realizado o fechamento da imagem gerando o resultado L_p , que consiste na imagem de entrada do segmentador de textos. Segue a operação matemática dessa etapa:

$$L_p = ((I \bullet S_r) \cap (I \bullet S_c)) \bullet S \quad (1)$$

Onde:

- S_r : Elemento estruturante de 1 pixel de altura e 100 pixels de largura;
- S_c : Elemento estruturante de 200 pixels de altura e 1 pixel de largura;
- S : Elemento estruturante de 1 pixel de altura e 30 pixels de largura;

A operação da Equação (1) tem como objetivo preparar a imagem para a segmentação de texto em geral, identificando as linhas do mesmo.

Também será realizado o pré-processamento da imagem de entrada a fim de segmentar as palavras do texto identificado. Para isso, é realizado uma operação de dilatação seguido de um fechamento, gerando a imagem W_p como mostrado a seguir:

$$W_p = (I \oplus S_d) \bullet S_f \quad (2)$$

Onde:

- S_d : Elemento estruturante de 6 pixels de altura e 1 pixel de largura;
- S_f : Elemento estruturante de 1 pixel de altura e 10 pixels de largura;

2.2 Segmentador de Textos

O segmentador de textos utiliza a imagem L_p obtida na etapa anterior como entrada.

Os passos realizados nessa etapa são:

1. Identificar os *bounding-box* das componentes conexas de L_p ;
2. Para cada bounding-box, calcular dois índices:
 - (a) Razão entre o número de pixels pretos e o número total de pixels, denominado R_a ;
 - (b) Razão entre o número de transições verticais e horizontais branco para preto e o número total de pixels pretos, denominado R_t ;

Um filtro será aplicado nesses dois índices de modo a separar as bounding-box compreendendo texto das que não compreendem texto, satisfazendo as condições:

$$0.5 < R_a < 0.9$$

$$0 < R_t < 0.1$$

2.3 Segmentador de Palavras

O segmentador de palavras utiliza a imagem W_p obtida na etapa de pré-processamento como entrada.

Nesta etapa, será identificado os *bounding-box* das componentes conexas de W_p e realizado uma intersecção com o resultado obtido no segmentador de texto. Dessa forma, teremos apenas as *bounding-box* de W_p dentro da região compreendida como texto.

3 Entrada de Dados

O script foi escrito utilizando *Python* 3.7, com as bibliotecas *scikit-image*, *OpenCV*, e *NumPy*.

Para a execução do programa basta utilizar o seguinte comando:

```
python3 segtext.py bitmap.pbm output_path
```

Sendo os parâmetros:

- *segtext.py*: Nome do script;
- *bitmap.pbm*: Caminho da imagem binária de entrada;
- *output_path*: Caminho do diretório onde será salvo as imagens de saída.

4 Detalhes de Implementação

Para a leitura e escrita das imagens foi utilizado as funções *imread* e *imwrite* do *OpenCV*, que suporta também o formato *pbm*.

Para as operações de morfologia matemática foi utilizado o pacote *morphology* do *scikit-image* [1], em especial as funções otimizadas para imagens binárias: *binary_closing* e *binary_dilation*.

O segmentador de palavras consiste na intersecção das componentes conexas do segmentador de texto com a imagem W_p , resultado do pré-processamento. Para realizar essa intersecção foi utilizado o seguinte algoritmo:

1. Transformar as *bounding-box* do segmentador de texto em uma máscara binária com 1 dentro das *bounding-box* e 0 fora;
2. Transformar as *bounding-box* da imagem W_p da mesma forma que o passo anterior;
3. Realizar a intersecção das duas máscaras, gerando a máscara de palavras;
4. Transformar a máscara de palavras em *bounding-box* com o algoritmo de componentes conexas.

5 Resultado e Discussão

O resultado do pré-processamento para segmentação de texto com os índices R_a e R_t calculados, respectivamente, na Figura 1. É possível perceber o padrão de valores dos índices nas linhas do texto que, após filtrado, resulta nas *bounding-box* da Figura 2

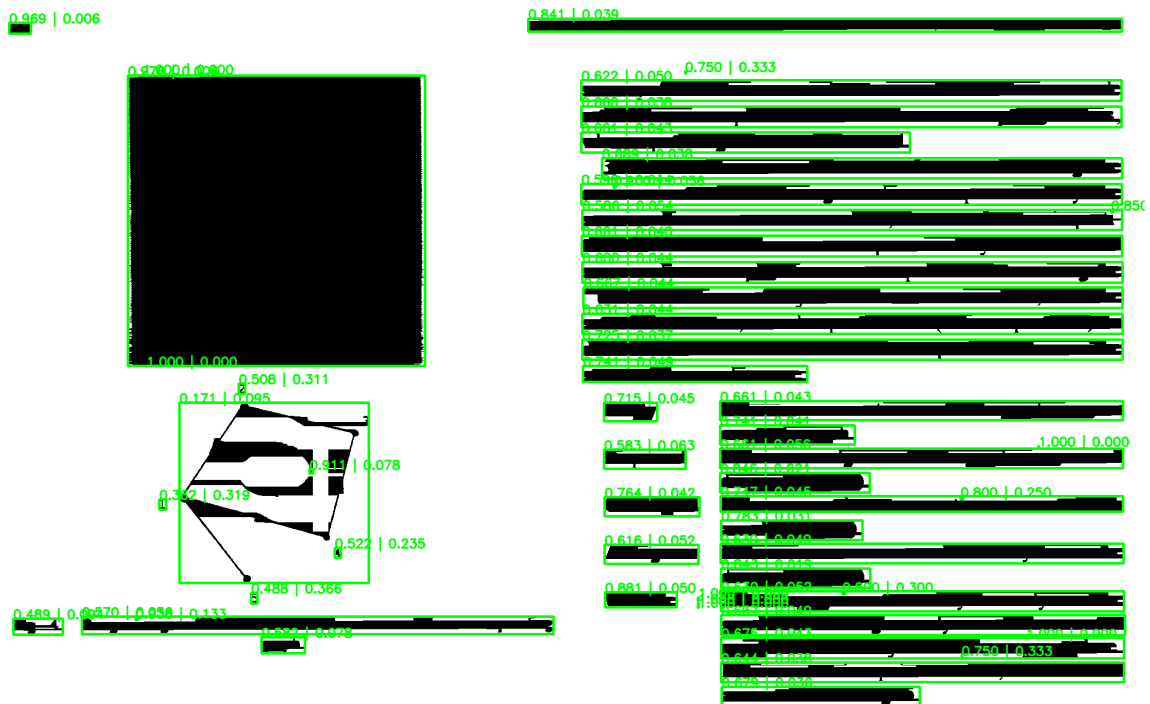


Figura 1: Índices calculados para o segmentador de texto

O pré-processamento para identificação das palavras resulta na Figura 3 e a segmentação de palavras na Figura 4.

o número de linhas do texto foi calculado contando as *bounding-box* da imagem 2 e a quantidade de palavras contando na Figura 4. O resultado encontrado foi 33 linhas e 234 palavras.

6 Conclusão

Neste trabalho foi explorado operações de morfologia matemática em imagens binárias para segmentação de texto. A partir de dois índices conseguimos segmentar a região de texto em uma página de artigo contendo imagem e gráfico.

Os próximos passos para aprimorar o algoritmo será testar em novas imagens e encontrar melhores *thresholds* dos índices. Para isso também pode-se utilizar algoritmos de aprendizado de máquina, desde que haja uma grande quantidade de imagens já segmentadas para a fase de treino e teste.

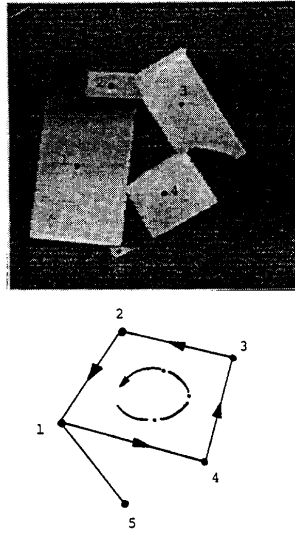


Fig. 4. Range image of an AMBIGUOUS scene and the corresponding graph.

sensory feedback is carried out in a local reflexive mode rather than in a planned mode with one exception, that is, when a pathological state is detected.

3) *States*: This is a finite set of states describing the environment of the Turing machine as perceived by the sensors. If new sensors are added, the set of states is partitioned to describe the scene as perceived by the additional sensors. For example, if a sensor capable of determining the "touch" relations of objects in the scene is added, then the set of five states, can be partitioned (a finer partition) to describe both the "touch" and "on-top-of" relations. The states of the machine are:

Empty	If there are no vertices in the diagraph, i.e., an empty diagraph.
Dispersed	If there no edges in the diagraph, i.e., a null diagraph (Fig. 2).
Overlapped	If there are at least two vertices connected with an edge (Fig. 3).
Ambiguous	If there is one or more directed cycles in the diagraph (Fig. 4).
Unstable	This category is not tested by the analysis of the graph but through analysis of the contact point/line of the object with the support plane. If this contact is a point or a line, it is classified as unstable. See Fig. 5.

Figura 2: Texto Segmentado

Referências

- [1] Scipy Morphology. Mathematical morphology operations. https://scikit-image.org/docs/dev/api/skimage.morphology.html#skimage.morphology.binary_dilation. [Online; acessado em 17 de Maio de 2019].

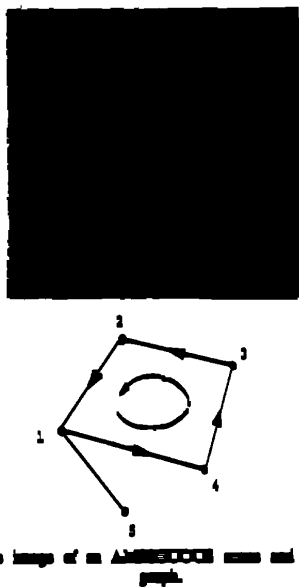


Fig. 4. Range image of an AMBIGUOUS scene and the corresponding graph.

sensory feedback is carried out in a local reflexive mode rather than in a planned mode with one exception, that is, when a pathological state is detected.

3) *States*: This is a finite set of states describing the environment of the Turing machine as perceived by the sensors. If new sensors are added, the set of states is partitioned to describe the scene as perceived by the additional sensors. For example, if a sensor capable of determining the "touch" relations of objects in the scene is added, then the set of five states, can be partitioned (a finer partition) to describe both the "touch" and "on-top-of" relations. The states of the machine are:

- Empty** If there are no vertices in the diagram, i.e., an empty diagram.
- Dispersed** If there are no edges in the diagram, i.e., a null diagram (Fig. 2).
- Overlapped** If there are at least two vertices connected with an edge (Fig. 3).
- Ambiguous** If there is one or more directed cycles in the diagram (Fig. 4).
- Unstable** This category is not tested by the analysis of the graph but through analysis of the contact point/line of the object with the support plane. If this contact is a point or a line, it is classified as unstable. See Fig. 5.

Figura 3: Pré-processamento para identificação de palavras

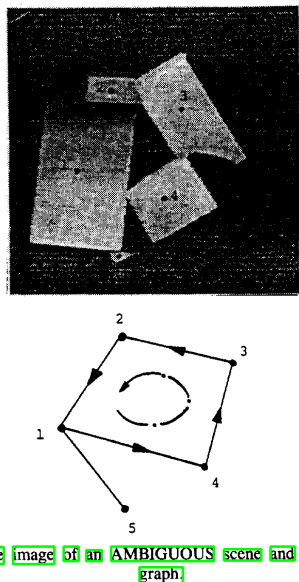


Fig. 4. Range image of an AMBIGUOUS scene and the corresponding graph.

sensory feedback is carried out in a local reflexive mode rather than in a planned mode with one exception, that is, when a pathological state is detected.

3) *States*: This is a finite set of states describing the environment of the Turing machine as perceived by the sensors. If new sensors are added, the set of states is partitioned to describe the scene as perceived by the additional sensors. For example, if a sensor capable of determining the "touch" relations of objects in the scene is added, then the set of five states, can be partitioned (a finer partition) to describe both the "touch" and "on-top-of" relations. The states of the machine are:

- Empty** If there are no vertices in the diagram, i.e., an empty diagram.
- Dispersed** If there are no edges in the diagram, i.e., a null diagram (Fig. 2).
- Overlapped** If there are at least two vertices connected with an edge (Fig. 3).
- Ambiguous** If there is one or more directed cycles in the diagram (Fig. 4).
- Unstable** This category is not tested by the analysis of the graph but through analysis of the contact point/line of the object with the support plane. If this contact is a point or a line, it is classified as unstable. See Fig. 5.

Figura 4: Palavras Segmentadas