UNIVERSIDADE FEDERAL DE MINAS GERAIS
Departamento de Ciência da Computação

Disciplina	Turmas
Introdução à Computação Visual	

Professores
Erickson Nascimento e William Robson Schwartz

Entrega: 04/07/2017 até às 23h59 (via moodle)

Trabalho Prático 3: Contagem de caracteres utilizando morfologia matemática

O objetivo deste trabalho é utilizar operações de morfologia matemática para contar o número de letras existentes em cada palavra de um texto, como o mostrado na Figura 1. *O trabalho deve ser feito individualmente.*

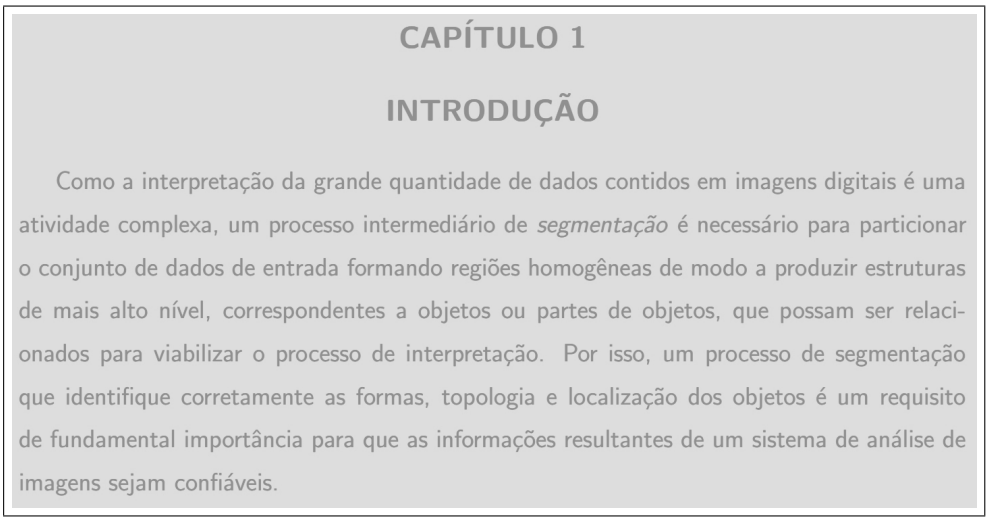

Figura 1: Exemplo de imagem de entrada.

O que deve ser feito

Como continuação do primeiro trabalho, cujo o objetivo era demarcar a localização de palavras em um texto, o objetivo deste trabalho é implementar um método para efetuar a contagem de letras que cada palavra localizada possui. A entrada será um fragmento de texto, como o mostrado na Figura 1 (note que a imagem está com baixo contraste). Caso alguma palavra no texto esteja separado por hífen (e.g., "relacionados", na imagem de exemplo), poderá ser considerada como sendo duas palavras distintas (neste exemplo, "relaci" e "onados" serão consideradas duas palavras com 6 letras cada). Nota importante: serão consideradas "palavras" somente agrupamentos de letras e/ou números. Toda pontuação deve ser descartada da imagem resultante final (e.g., ponto final, vírgula, hífen, etc.) e toda acentuação deve ser mantida (e.g., acento, til, pingo do i, etc.).

Os algoritmos deste trabalho deverão ser implementado utilizando linguagem Python, com a utilização da biblioteca OpenCV versão 3, dentro do ambiente *Jupyter Notebook*[1]. Também podem ser

[1] `http://jupyter.org/`

utilizadas bibliotecas já instaladas, mas não podem ser utilizadas bibliotecas especificas que requerem instalação adicional.

As operações morfológicas a serem utilizadas (erosão, dilatação, etc.) deverão **ser implementadas**. A escolha dos elementos estruturantes adequados em cada etapa é parte do trabalho. Para selecionar os elementos estruturantes, será permitida a utilização da função "cv2.getStructuringElement", do OpenCV (https://goo.gl/YJEouB). As operações para melhorar o contraste da imagem também deve ser implementadas.

O método deve gerar as seguintes saídas:

1. Colocar um retângulo entorno de cada palavra do texto, como já realizado no primeiro trabalho prático e mostrado na Figura 2. Você poderá utilizar o código já implementado no primeiro trabalho (sem penalidades na nota deste trabalho). Dica: como alguns caracteres ficam fora das bordas demarcadas (conforme ilustrado na Figura 2) recomenda-se fazer esta operação utilizando morfologia matemática.

2. Gerar um arquivo de texto (.txt) contendo as coordenadas do retângulo contendo cada palavra localizada no texto, uma por linha e o número de letras que esta palavra tem. Cada linha do arquivo texto deve ter o formato: `x, y, w, h, n`, onde (x,y) indica a localização do canto superior esquerdo da palavra, w, h, indicam a largura e a altura do retângulo entorno da palavra e n indica o número de letras encontrados para aquela palavra, respectivamente.

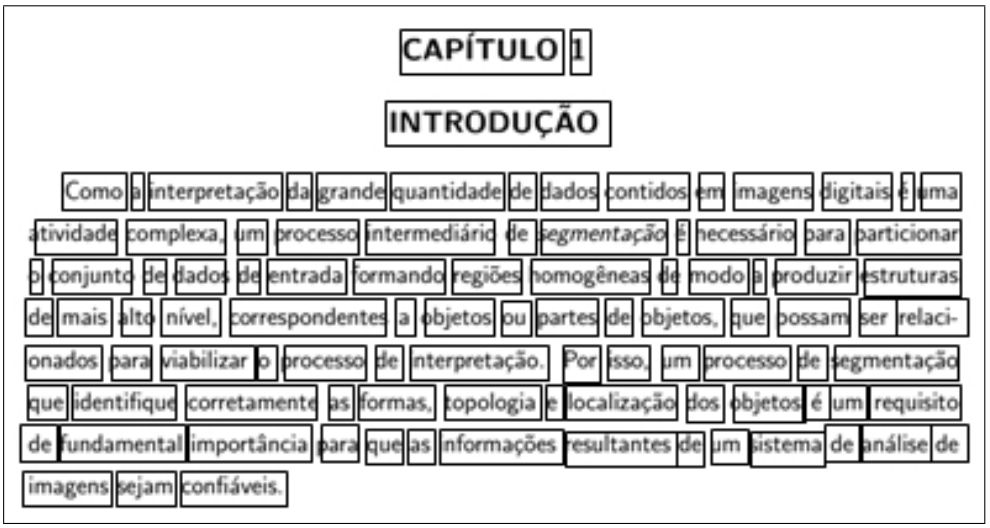

Figura 2: Resultado do trabalho prático #1.

Modularize o seu código criando uma (ou mais) célula do tipo *Code* para cada etapa do método. Por exemplo, crie uma célula para definir a função de erosão, outra para definir a função de dilatação, uma para definir as funções de abertura e fechamento (chamando as duas anteriores), outra para corrigir o contraste da imagem, outra para efetuar a limiarização da imagem, outra para determinar a posição das palavras e assim por diante. Ao final de cada etapa, se pertinente, apresente resultados intermediários (isso facilita na correção e na compreensão do método). Para cada bloco do código, crie células do tipo *Markdown* explicando os passos e as escolhas feitas para o método (por exemplo, escolha dos elementos estruturantes).

O que deve ser entregue

O notebook salvo no ambiente Jupyter deve ser submetido no Moodle.

Execução

O notebook submetido será carregado no Jupyter e outra imagem (com as mesmas características das imagens providas, mas possivelmente com texto em outras localizações) será carregada e o código será executado passo a passo para efetuar a correção. Portanto, é importante que as etapas do método sejam implementadas em células com visualização dos resultados intermediários, juntamente com as devidas explicações em comentários e em células do tipo *Markdown*.