

Relatório - Projeto 1

Caio Augusto Alves Nolasco - RA:195181

MC940 - Processamento e Análise de Imagens - 1º Semestre 2020

1. Introdução

O código criado referente ao projeto 1 tem, de forma geral, realizar alterações sobre os valores de níveis de cinza de cada pixel de uma imagem, baseadas em transformações matemáticas com um propósito bem definido. No caso, são feitas transformações como a quantização dos níveis de cinza de uma imagem, alterando o valor máximo possível para todos os pixels. Também são realizadas operações matemáticas sobre os valores de níveis de cinza, como aplicar a função exponencial para cada elemento da matriz de pixels.

O objetivo é alcançar modificar as imagem de determinada maneira alterando apenas os valores em sua representação matricial.

O programa é dividido em uma função para cada transformação proposta pelo enunciado do projeto. São feitas algumas suposições sobre os parâmetros pertencentes aos problemas. As suposições comuns a todas as transformações são:

- A imagem dada como entrada está no formato PNG.
- A imagem dada como entrada é aberta via a biblioteca PIL em forma de imagem PIL ou pela biblioteca ImageIO como um array de *numpy*, dependendo de qual alternativa é melhor adequada para determinada transformação.
- A imagem dada como entrada tem tamanho 512x512, e este fato é considerado como sabido na resolução do problema
- O caminho do arquivo da imagem é fornecido em *runtime* pelo usuário
- O caminho do arquivo de saída desejado é fornecido pelo usuário
- A imagem dada como entrada tem como valor máximo de nível de cinza igual a 255, e este fato é conhecido pelo programa
- As bibliotecas usadas são imageio, PIL, numpy e math.

Questão 1.1) Resolução de imagens

A função de alteração de resolução de uma imagem recebe como parâmetro um objeto de imagem PIL. Para realizar a mudança, o conteúdo da imagem em si não é alterado, mas sim a metadata referente ao dpi da imagem, usando um parâmetro opcional da função `save()` do módulo `PIL.Image`. São salvas quatro novas imagens, com resoluções

de 256x256, 128x128, 64x64 e 32x32, cada uma com o nome 'resolutionX.png', em que X representa o valor da resolução do arquivo específico.

Questão 1.2) Quantização de imagens

A função de quantização de uma imagem recebe como parâmetro uma imagem aberta como objeto de PIL. O módulo PIL.Image dispõe de uma função `quantize()`, que recebe como parâmetro o novo limite máximo para os níveis de cinza. A resolução faz quatro chamadas do método de PIL.Image, inicialmente com o valor original de 256, e outras três com valores de 128, 64, 32 e 16, e salva cinco novas imagens com os nomes 'quantized_imageX.png', em que X representa o parâmetro usado.

Questão 1.3) Escala de cinza

Para as transformações matemáticas das escalas de cinza, a imagem é representada em uma matriz *NumPy*, e cada elemento $f(x,y)$ da matriz é substituído por seu correspondente a uma função $g(x,y) = c * h(f(x,y))$, em que $h(x,y)$ é especificada para cada caso, e c é uma constante de correção definida a partir de $h(x,y)$ para realçar a disparidade dentre os novos valores da matriz e adequá-los a escala original.

Cada transformação recebe um vetor `numpy_image` que representa a imagem original na forma matricial, e escreve sua saída em uma imagem PNG com o mesmo nome da transformação.

- a) Função logarítmica: substituí cada valor por $g(x,y) = c * \log_{10}(f(x,y) + 1)$. Para este caso, $c = 255/\log_{10}(\text{valor_max}() + 1)$.
- b) Função exponencial: substituí cada valor por $g(x,y) = c * \exp(f(x,y))$. Para este caso, $c = 1/\exp(\text{valor_max}())$. Antes da transformação, os valores são divididos por 255 para evitar overflow.
- c) Função quadrática: substituí cada valor por $g(x,y) = c * (f(x,y))^2$. Para este caso, $c = 255^2 / \text{valor_max()}^2$.
- d) Função raiz quadrada: substituí cada valor por $g(x,y) = c * (f(x,y))^{(1/2)}$. Neste caso, $c = 255/(\text{valor_max()}^{(1/2)})$.
- e) Alargamento de contraste:

$$g(x,y) = \begin{cases} \text{alfa} * f(x,y) , & \text{se } f(x,y) < a \text{ (I)} \\ \text{beta} * (f(x,y) - a) + \text{alfa} * a, & \text{se } f(x,y) > a \text{ e } f(x,y) < b \text{ (II)} \\ \text{gama} * (f(x,y) - b) + \text{beta} * (b - a) + \text{alfa} * a, & \text{se } f(x,y) > b \text{ e } f(x,y) < L = 255 \text{ (III)} \end{cases}$$

Para este item, os valores de a , b , alfa , beta e gama são considerados parâmetros entrados pelo usuário em tempo de execução.

Para realizar a operação, é criada uma matriz de zeros 512x512. Usando a função `where()` de NumPy, primeiro todos os números de satisfazem a condição (I) são alterados para o novo valor e os que não atendem a condição são zerados. Sucessivamente, o mesmo é feito para para os valores que atendem as condições (II) e (III), e o resultado final provém da soma dos três resultados intermediários.