

MO443 - Processamento de Imagem Digital

Trabalho 01

Patrick de Carvalho Tavares Rezende Ferreira - 175480

April 2020

1 Introdução

O objetivo deste trabalho é realizar alguns processamentos básicos em imagens digitais. Será empregada, sempre que pertinente, a vetorização dos comandos.

As operações requisitadas são as seguintes:

- Alteração da resolução de uma imagem;
- Quantização de imagens, alterando a quantidade de valores possíveis para representar a intensidade de cada pixel;
- Operações matemáticas sobre a escala de cinza de cada imagem, operando funções contínuas e contínuas por partes que serão descritas adiante.

2 Descrição do código elaborado

O código desenvolvido para este projeto possui uma função para a resolução de cada um dos 3 itens pedidos no roteiro. O código contido no arquivo "main.py" foi desenvolvido para que o utilizador possa simplesmente importar todas as suas funções e utilizá-las independentemente. Para fins de demonstração, o código também possui um script dentro da função "main()" que é executado sempre que chamado como arquivo principal pelo interpretador python e cria um diretório "output" contendo as imagens solicitadas neste roteiro. Todas as funções com argumentos numéricos aceitam que estes sejam inteiros ou pontos flutuantes positivos, tratando de forma adequada e convertendo para o tipo "unit8" a matriz da imagem antes de salvar o arquivo de saída.

2.1 Executando o script de demonstração

Dada uma imagem original, o roteiro solicita que sejam geradas imagens consecutivas metade da resolução de cada imagem anterior e imagens com o nível de quantização sucessivamente dividido por dois. Além disso, pede-se que implementemos uma lista de funções matemáticas a serem operadas, dada uma imagem original. Todas estas imagens são geradas quando se executa o script "main.py" como arquivo principal. Um exemplo de como executar o script neste modo sob a linha de comando é exibido na figura 1. A imagem usada como exemplo é a "baboon.png" e deve estar no mesmo diretório de chamada do script.

```
python main.py
```

Figura 1: Comando para gerar as saídas requisitadas pelo roteiro.

Todas as imagens solicitadas pelo roteiro serão então geradas em um diretório de "output" automaticamente criado no mesmo diretório onde o script é chamado. Os nomes de cada arquivo indicam a resolução de cada imagem, o nível de quantização ou a operação matemática que foi realizada na mesma, conforme o solicitado. A figura 3 exibe os arquivos de saída esperados.

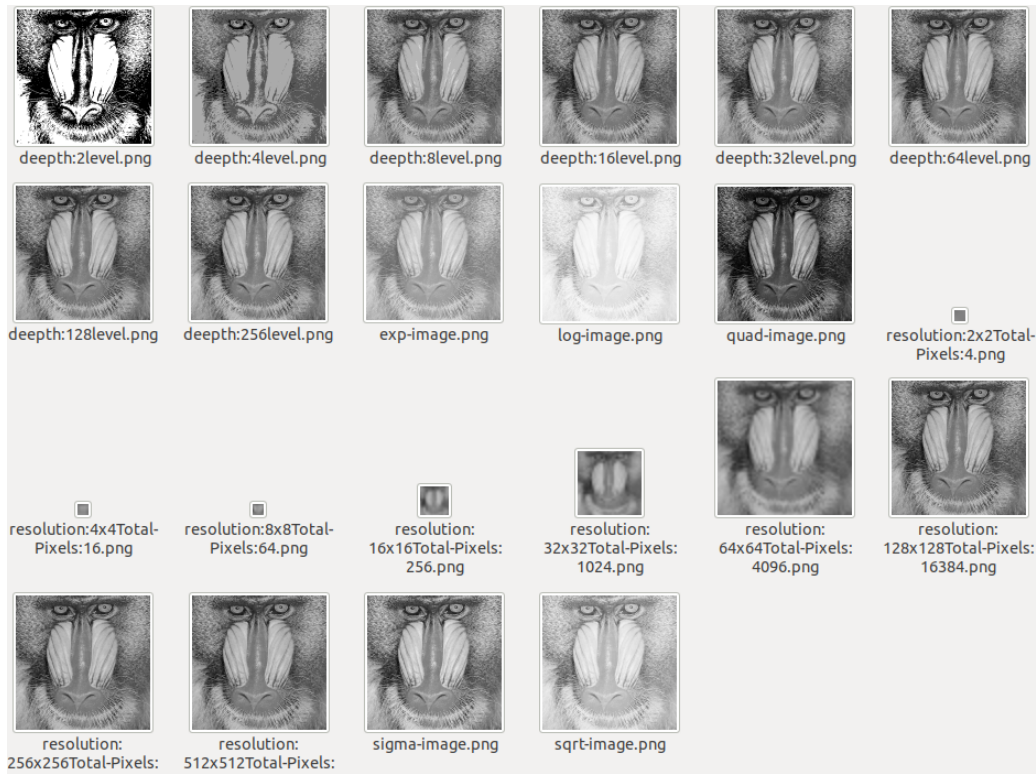


Figura 2: Imagens de saída.

2.2 Utilizando as funções interativamente

A forma com que o script foi construído nos permite importar suas funções e utilizá-las independentemente. Para fazê-lo, basta realizar o *import* tradicional do python, como na figura ??.

A demonstração da utilização de cada função é descrita a seguir.

2.2.1 Resolução de imagens

Para alterar a resolução de uma imagem em escala de cinza com pixels variando no valor de 0 a 255, foi desenvolvida a função "change resolution". Seu "header" é apresentado na figura 4 e seus

```

$ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from main import *
>>>

```

Figura 3: Importando funções desenvolvidas para este roteiro.

parâmetros detalhados na tabela 1.

```

def change_resolution(original_file_path, output_file_path="resized-img.png", scale_percent=100):
    """
    Receives an grayscale image path and saves it back with new resolution scale.
    :param original_file_path: Original grayscale uint8 image.
    :param output_file_path: Rescaled output grayscale uint8 image.
    :param scale_percent: Reduction rate
    :return: 0 if success.
    """

```

Figura 4: Função para editar a resolução.

Tabela 1: Referência da função change resolution.

Parâmetro	Descrição
original file path	Caminho absoluto para a imagem que se deseja tratar
output file path	Caminho absoluto para o arquivo de imagem de saída (default: resized-img.png)
scale percent	O percentual do tamanho que se deve manter para cada eixo
Retorno	Retorna 0 se a operação for bem sucedida e código de erro caso contrário

2.2.2 Quantização de imagens

A quantização de uma imagem é a quantidade de valores possíveis para cada pixel assumir em um dado range de intensidade. Para alterar a quantização de uma imagem em escala de cinza com pixels variando no valor de 0 a 255, foi desenvolvida a função "change deepth". Seu "header" é apresentado na figura 5 e seus parâmetros detalhados na tabela 1.

2.2.3 Escala de cinza

Para alterar os níveis de cinza de uma imagem, foram propostas uma série de transformações contínuas e contínuas por partes. As transformações implementadas sobre cada pixel de entrada (input) e de saída (output) são descritas a seguir:

- logaritmo: $\text{output} = \log(\text{input} + 1)$
- exponencial: $\text{output} = \exp(\text{input})$
- quadrado: $\text{output} = \text{input} * \text{input}$
- raiz quadrada: $\text{sqrt}(\text{input})$

```
def change_depth(original_file_path, output_file_path="depth-img.png", original_depth=256, new_depth=256):
    """
    Changes the depth for a given grayscale uint8 image.
    :param original_file_path: Original grayscale uint8 image.
    :param output_file_path: New depth output grayscale uint8 image.
    :param original_depth: Original image depth.
    :param new_depth: New image depth
    :return: 0 if success.
    """
```

Figura 5: Função para editar a resolução.

Tabela 2: Referência da função change depth.

Parâmetro	Descrição
original file path	Caminho absoluto para a imagem que se deseja tratar
output file path	Caminho absoluto para o arquivo de imagem de saída (default: depth-img.png)
original depth	A quantização inicial da imagem (default: 256 níveis)
new depth	A nova quantização da imagem (default: 256 níveis)
Retorno	Retorna 0 se a operação for bem sucedida e código de erro caso contrário

- alargamento de contraste:
$$\begin{cases} \alpha * input, & \text{se } 0 \leq input < a; \\ \beta * (input - a) + \alpha * a, & \text{se } a \leq input < b; \\ \gamma * (input - b) + \beta * (b - a) + \alpha * a, & \text{se } b \leq input < L. \end{cases}$$

Para realizar estas operações sobre uma imagem em escala de cinza com pixels variando no valor de 0 a 255, foi desenvolvida a função "change contrast". Seu "header" é apresentado na figura 6 e seus parâmetros detalhados na tabela 3.

```
def change_contrast(original_file_path, output_file_path="contrast-img.png", operation=None, alpha=2, beta=2, gamma=2):
    """
    Manipulates contrast on grayscale uint8 image.
    :param original_file_path: Original grayscale uint8 image.
    :param output_file_path: New depth output grayscale uint8 image.
    :param operation: Which operation to perform as string (log, exp, quad, sqrt or sigma).
    :param alpha: alpha factor. Default is 2.
    :param beta: beta factor. Default is 2.
    :param gamma: gamma factor. Default is 2.
    :return: 0 if success.
    """
```

Figura 6: Função para alargamento de contraste.

A operação determinada pelo argumento "operation" é determinada, respectivamente em relação às transformações solicitadas, pelas strings 'log', 'exp', 'quad', 'sqrt' ou 'sigma' (sigma representa o alargamento de contraste e o nome se deve à vaga semelhança gráfica com uma curva sigmoide). Como as operações fazem com que os valores dos pixels se tornem muito maiores ou menores que o range de 0 a 255 padrão, todas elas são interpoladas ao final da operação, de forma que seu valor máximo se alinhe a 255 e o valor mínimo a 0. Ademais, especificamente no caso da operação exponencial, a única forma encontrada de realizá-la sem promover discrepância entre os valores dos pixels foi normalizar a imagem para o intervalo [0, 1] antes de realizar a operação, interpolando-a novamente para [0, 255] em seguida. Os valores dos parâmetros fixos a, b e c foram definidos como 255/3, 2*255/3 e 1, respectivamente, de forma que a e b dividissem os valores dos pixels em 3 áreas iguais e c não alterasse a escala de cada operação, para não afetar a aparência do resultado.

Tabela 3: Referência da função change contrast.

Parâmetro	Descrição
original file path	Caminho absoluto para a imagem que se deseja tratar
output file path	Caminho absoluto para o arquivo de imagem de saída (default: depth-img.png)
operation	A operação a ser desempenhada sobre os pixels.
alpha, beta e gamma	Parâmetros a serem utilizados caso se desempenhe o alargamento de contraste
Retorno	Retorna 0 se a operação for bem sucedida e código de erro caso contrário

3 Efeito de cada uma das operações implementadas

As imagens de saída solicitadas pelo roteiro são representadas na figura 3, as quais são debatidas aqui. O efeito de se mudar a resolução já é conhecido pela maioria das pessoas e implica em alterar a riqueza de detalhes que se pode distinguir em casa imagem, visto que cada pixel fica responsável por descrever uma área maior da imagem conforme a resolução diminui. Reduzir o nível de quantização da imagem faz com que a variação entre os valores dos pixels seja mais abrupta, como era de se esperar, resultando em imagens com aspecto mais granular conforme se diminui a quantização.

Quanto às operações matemáticas, os resultados específicos para cada uma. A operação de log parece diminuir o contraste, ao passo que aproxima todos os pixels para valores mais baixos de intensidade, sendo que a variação é maior em função de quão alto era o valor do pixel antes da transformação. A exponenciação faz com que pixels de maior brilho se destaquem mais que os outros, clareando ainda mais os pixels mais claros e tornando maior a discrepância entre pixels de diferentes valores. A operação de potenciação ao quadrado, assim como a exponencial, acentua a disparidade entre pixels de diferentes valores, porém em menor intensidade do que a exponencial, tornando a imagem em regiões muito claras (pixels de alto valor elevados ao quadrado) e outras muito escuras (pixels de valores mais baixos em relação aos demais e que acabam não se ressaltando). A operação de raiz quadrada (sqrt) também torna os pixels de maior valor em pontos mais próximos daqueles que possuíam baixos valores antes da transformação, de forma semelhante ao logaritmo, porém em menor intensidade, dando à imagem um aspecto "pálido".

Por último, a transformação de alargamento de contraste (sigma) aplica uma transformação diferente para cada faixa de valores iniciais do pixel, dando um efeito visual na imagem que parece aumentar a nitidez, aumentando de forma linear a diferença a intensidade de pixels de alto, médio e baixo valor. Isto tem o potencial de ressaltar linhas e bordas, tornando a imagem com curvas mais definidas.