

Mc920 - Trabalho 0

Felipe Santana Dias - 215775

1. ESPECIFICAÇÃO DOS PROBLEMAS

O objetivo do Trabalho 0 é gerar algumas transformações simples em imagens na escala de cinza, sendo elas:

1. Transformação de Intensidade, composta pelo negativo da imagem, espelhamento vertical, imagem transformada, linhas pares invertidas, reflexão de linhas;
2. Ajuste de brilho, de acordo com dado valor de gama;
3. Plano de Bits, de acordo com dado valor do plano (0 a 7 bits);
4. Mosaico;
5. Combinação de imagens, a partir de duas imagens.

2. ENTRADA DE DADOS

O código desenvolvido para resolver todas as questões levantadas no tópico 1 foram desenvolvidos em python através das bibliotecas OpenCV e NumPy em um arquivo chamado `'mc920-trabalho0.py'`. O programa foi feito para aceitar imagens na escala de cinza e no formato PNG.

Ao executar o código, será requerido o caminho da imagem que será utilizada como base, sendo necessário inserir o caminho da imagem desejada. As imagens testadas durante a criação do código são as disponibilizadas pelo professor e estão presentes na pasta raiz juntamente com o código.

Após escolhida a imagem, um menu numérico é apresentado para a escolha de quais das transformações deseja realizar. A numeração é correspondente ao especificado no tópico 1. Dada a escolha poderão ser requisitada novas informações específicas de cada transformação.

3. CÓDIGO E DECISÕES TOMADAS

3.1 Pré processamento

A imagem foi carregada através da biblioteca OpenCV e convertida em uma NumPy Array para realizarmos as operações de maneira otimizada.

3.2 Transformação de Intensidade

Cada operação realizada dentro da transformação de intensidade foi desenvolvida em funções separadas e completamente compostas por funções disponíveis na biblioteca NumPy e OpenCV visando a otimização no tempo de processamento.

3.3 Ajuste de brilho

Sendo solicitado inicialmente um valor para gama, devendo este ser escrito na forma decimal. Essa função foi desenvolvida utilizando completamente funções das bibliotecas NumPy e OpenCV visando a otimização de processamento.

3.4 Plano de Bits

Sendo solicitado inicialmente um valor para o plano de bits, devendo este ser um inteiro no intervalo de 0 a 7. Com um tempo de processamento não ideal, os valores da imagem são convertidos para binário através da função *binary_repr* presente em NumPy e, posteriormente, é percorrido cada elementos selecionando a camada de bits correspondente.

3.5 Mosaico

Com um tempo de processamento não ideal, a imagem é dividida em 16 partes de dimensões 128px x 128px. Esses quadrantes são então rearranjados de acordo com a ordem pré-estabelecida no enunciado através de laços encaixados percorrendo cada quadrante. Por fim, a imagem é recomposta e finalizada com um *reshape* na sua forma original de 512px x 512px.

3.6 Combinação de Imagens

Sendo solicitado inicialmente o endereço de uma segunda imagem, é requisitado também o peso para a realização da média ponderada entre as imagens, devendo estes escritos na forma decimal (0.x). Essa função foi desenvolvida utilizando completamente funções das bibliotecas NumPy e OpenCV visando a otimização de processamento.

4. SAÍDA DE DADOS

Ao final de cada operação é gerada uma nova imagem cujo título é o mesmo que o inicial adicionado do nome da transformação. Todas as imagens são salvas na pasta raiz.

5. RESULTADOS

Os resultados foram satisfatórios e compatíveis com os presentes no enunciado, sendo seu tempo de processamento otimizado na maioria das funções.