

Exercício Visão Computacional

1) Implemente uma função que altere o brilho e/ou o contraste de uma imagem colorida. A função deve ter como entrada:

- uma imagem colorida ou preta e branca, com valores r, g, b no intervalo $[0,1]$
- o valor da alteração do brilho $\beta \in [-1,1]$
- o valor da alteração do contraste $\kappa \in [0,1]$

O brilho deve ser alterado de forma aditiva, isto é, o pixel com cor $c = (r, g, b)$ deve passar para $c = (r + \beta, g + \beta, b + \beta)$ (é a mesma cor, mas menor saturação – foi misturada com um pouco de cor branca).

O contraste deve ser alterado de forma multiplicativa, isto é, o pixel com cor $c = (r, g, b)$ deve passar para $c = (\kappa(r - \bar{r}) + \bar{r}, \kappa(g - \bar{g}) + \bar{g}, \kappa(b - \bar{b}) + \bar{b})$.

Teste com a imagem PoucoContraste.png ou outra imagem de sua preferência.

2) Faça o histograma da imagem original, a imagem com brilho alterado e a imagem com contraste alterado usando a função desenvolvida em (1). Compare com o histograma da imagem com brilho e contraste alterado da função

`cv2.convertScaleAbs(img, alpha=contraste, beta=brilho)`

A sua função implementa o mesmo conceito que a função do OpenCV?

3) Escreva uma função para efetuar a convolução de uma imagem com um filtro. A função deve ter como entrada:

- Uma imagem em preto e branco
- A matriz correspondente ao filtro

A saída deve ser a imagem filtrada. Teste com os seguintes filtros:

- 1) Constante 3x3
- 2) Derivada na horizontal 1x3 e derivada na vertical 3x1
- 3) Filtro de Sobel horizontal e vertical 3x3 (que tal olhar o módulo do gradiente também?)
- 4) Filtro gaussiano com média zero e variância de 2 pixels, truncado numa matriz 5x5.

As imagens para teste são à sua escolha, mas um dos testes deve ser com uma imagem de um tabuleiro de xadrez.

4) Faça a redução de uma imagem pela metade do número de linhas e metade do número de colunas de duas formas:

- a) Sem fazer uma suavização, apenas cortando linhas e colunas
- b) Com a suavização antes de eliminar linhas e colunas.

Você pode usar as funções do OpenCV para efetuar a suavização.