

**UNIVERSIDADE FEDERAL DO RIO
GRANDE DO SUL
INSTITUTO DE INFORMÁTICA**
INF01046 - Fundamentos de Processamento de
Imagens - 2017_2
Professor Manuel Menezes de Oliveira Neto

Trabalho 2 - Relatório

Parte 1

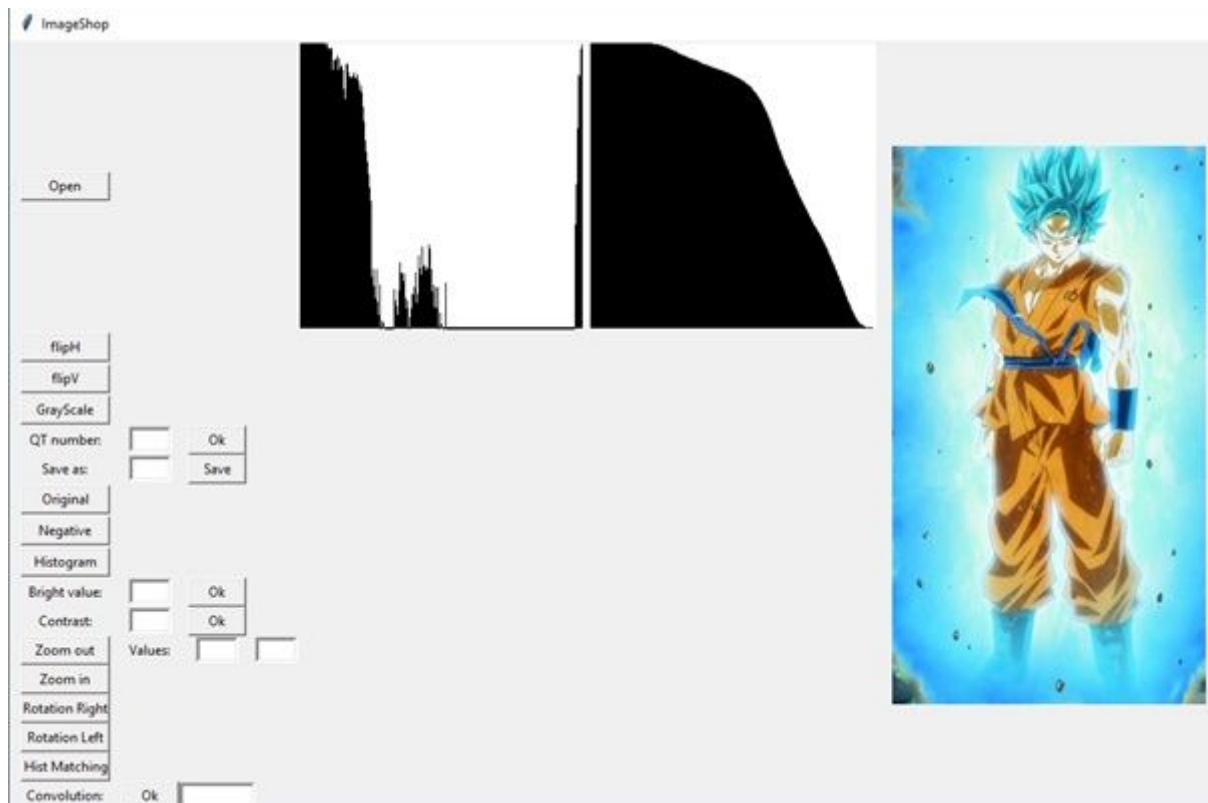
1) Calcular e exibir o histograma de uma imagem em tons de cinza (8 bits por pixel).

Caso a imagem informada como entrada seja colorida, converta-a para tons de cinza

(luminância) e então calcule seu histograma. Exiba o histograma em uma janela de 256x256 pixels, onde cada coluna da imagem representa um tom de cinza.

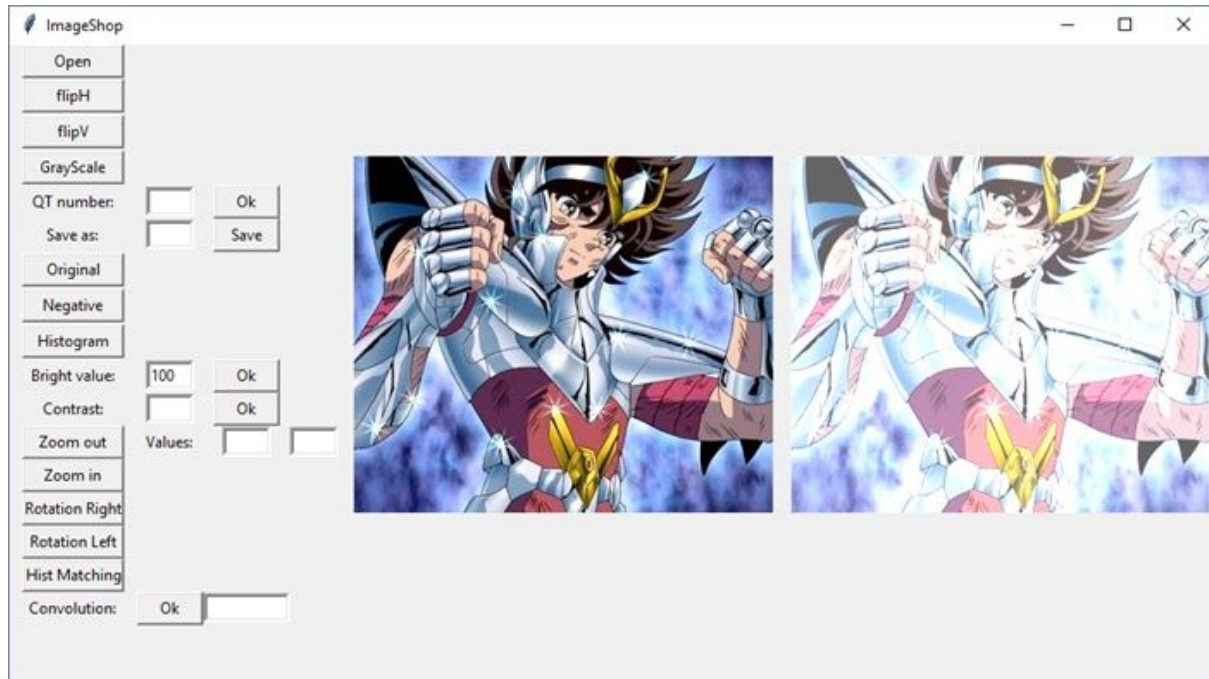
Normalize a altura das colunas para obter uma representação apropriada (20 pontos).

O histograma é calculado em um vetor de 255 posições, em que cada posição é um tom de cinza. Cada posição contém o número de vezes que o tom aparece na imagem. Na foto, temos o histograma normal e o histograma normalizado para a imagem teste.



2) Ajustar o brilho de uma imagem (e exibí-la), somando ao valor de cada pixel um escalar no intervalo $[-255, 255]$. Certifique-se que o resultado da operação aplicado a cada pixel encontra-se na faixa $[0, 255]$, ajustando-o para zero ou 255 quando necessário. No caso de imagens coloridas, aplique o algoritmo para cada um dos canais (R,G,B) independentemente (15 pontos).

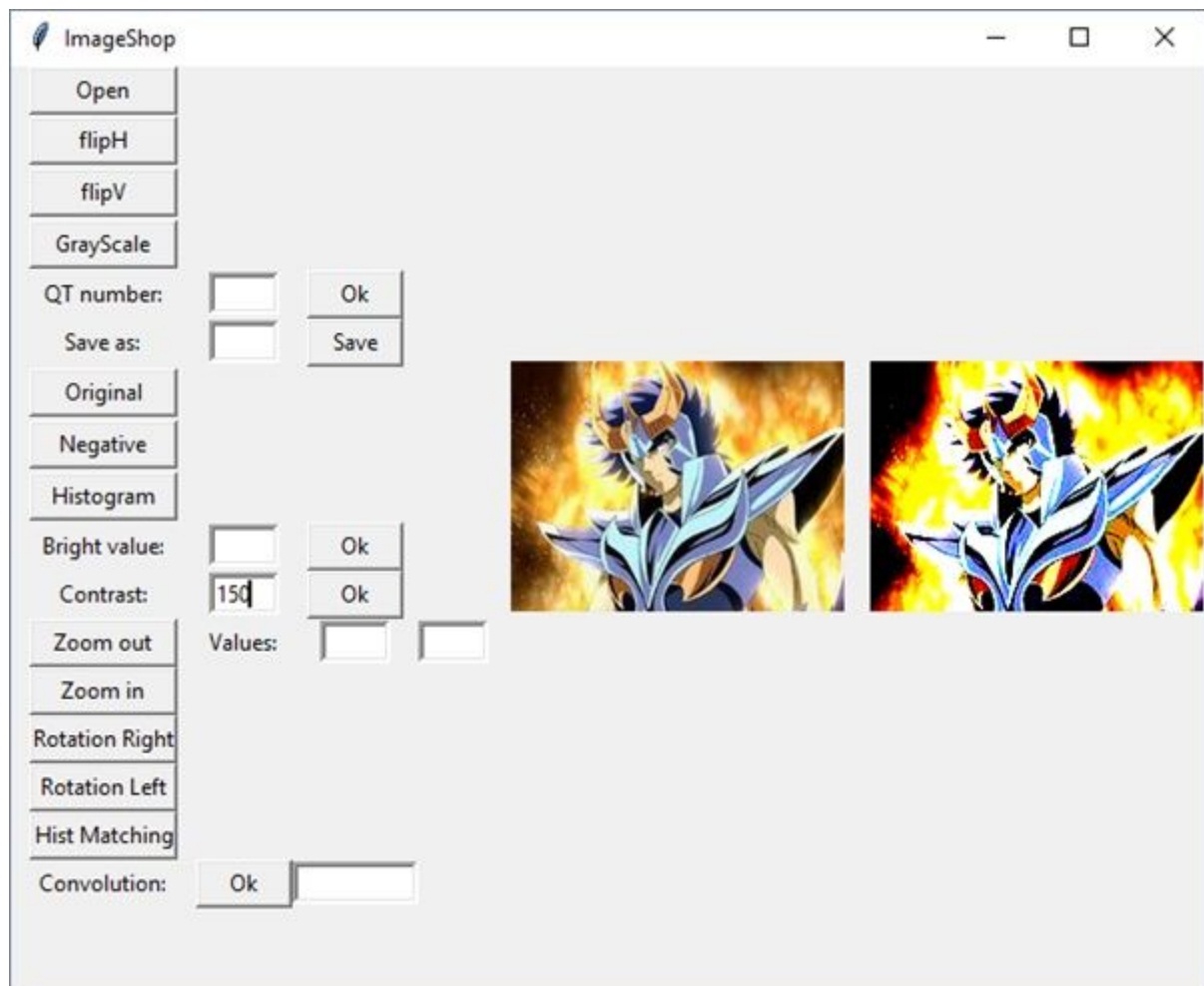
O brilho é calculado a partir do escalar dado como entrada. No caso do exemplo da imagem, o escalar dado foi 100, e portanto, o brilho da imagem está muito maior. Cada canal tem o escalar somado a seu tom, limitando para o máximo de 255.



3) Ajustar o contraste de uma imagem (e exibí-la), multiplicando cada pixel por um escalar no intervalo (0, 255]. Certifique-se que o resultado da operação aplicado a cada pixel encontra-se na faixa [0,255], ajustando-o para 255 quando necessário. No caso de imagens coloridas, aplique o algoritmo para cada um dos canais (R,G,B) independentemente (15 pontos).

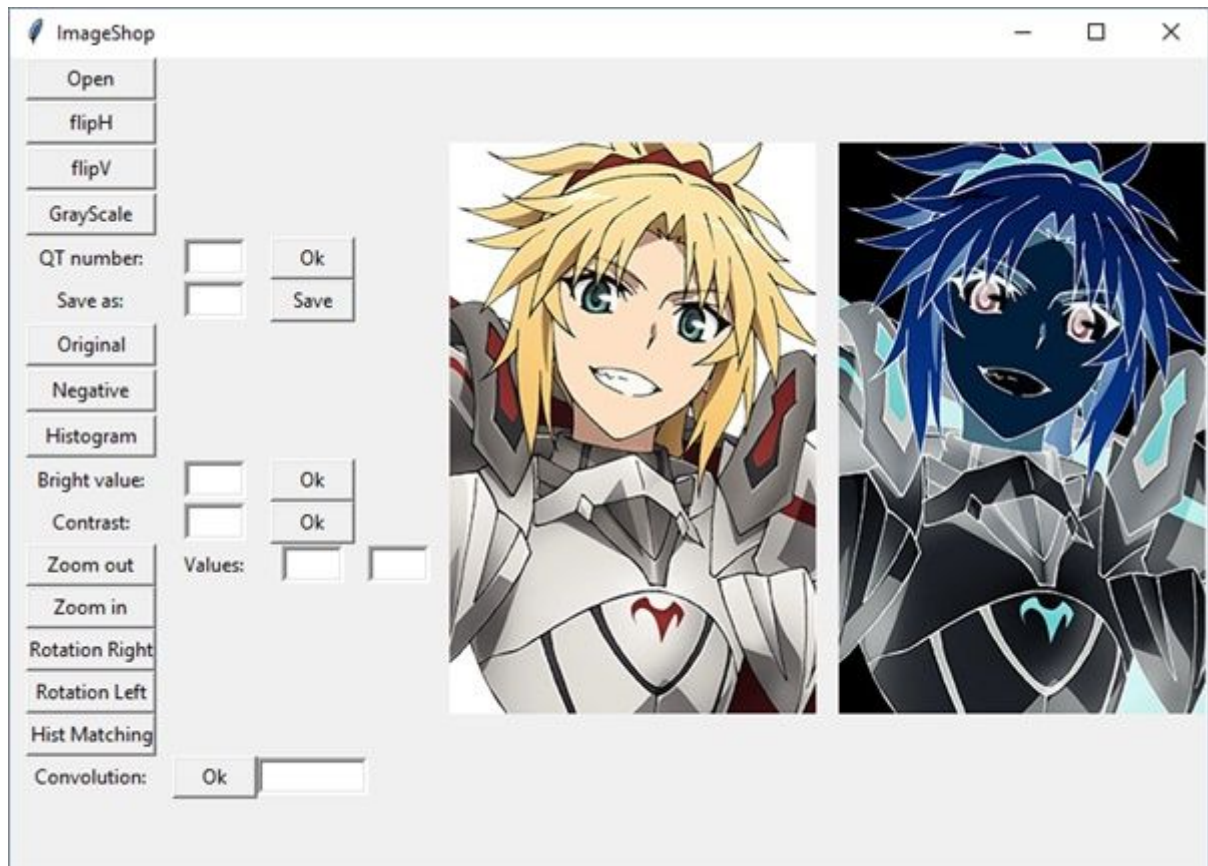
Para ajustar o contraste da imagem, similarmente ao brilho, o usuário dará um escalar que será usado para encontrar o Fator de contraste.

$$F = \frac{259(C + 255)}{255(259 - C)}$$
 Depois disso, para calcular a nova cor no canal, basta aplicar a seguinte fórmula: $R' = F(R - 128) + 128$
Desse modo, o contraste é ajustado com precisão.



4) Calcular e exibir o negativo de uma imagem, calculando o novo valor de cada pixel como: $= 255 -$. No caso de imagens coloridas, aplique o algoritmo para cada um dos canais (R,G,B) independentemente (10 pontos).

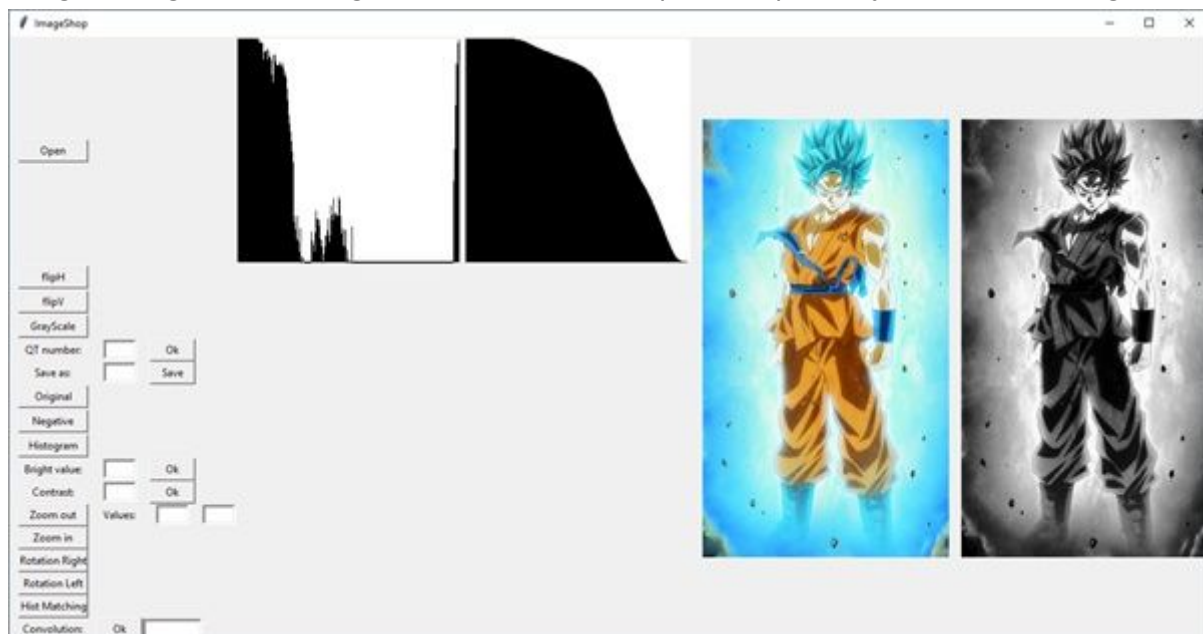
Simplesmente pegar o valor de cada canal do pixel e subtrair de 255.



5) Equalizar o histograma de uma imagem, exibindo as imagens antes e depois da equalização. Para imagens em tons de cinza, exibir os histogramas antes e depois da equalização. No caso de imagens coloridas, para cada um dos canais (R,G,B), utilize o histograma cumulativo obtido a partir da imagem de luminância (25 pontos).

Para o cálculo do histograma equalizado, utilizou-se a fórmula fornecida pelo professor, que consiste em calcular um $\text{scaling} = 255/(\text{número de pixels da imagem})$, obtendo-se o histograma cumulativo. Assim, o histograma equalizado é simplesmente o histograma cumulativo na posição anterior somado do scaling multiplicado pelo histograma.

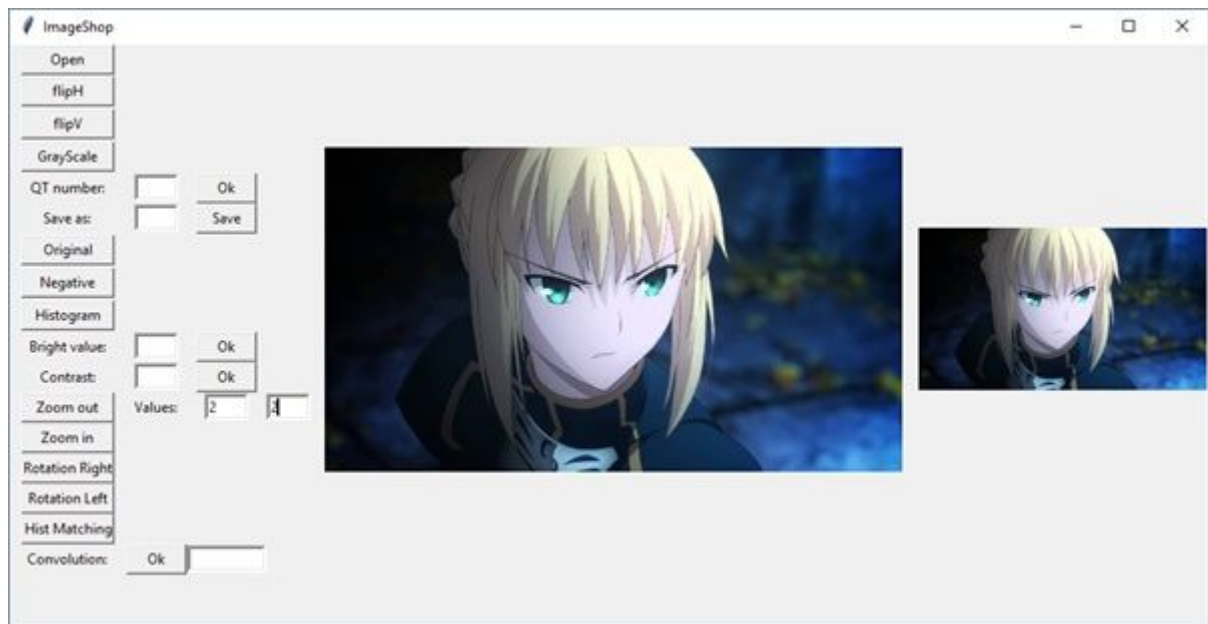
No exemplo, temos o histograma normal, o histograma equalizado, a imagem original e a imagem em luminância após a equalização de seu histograma.



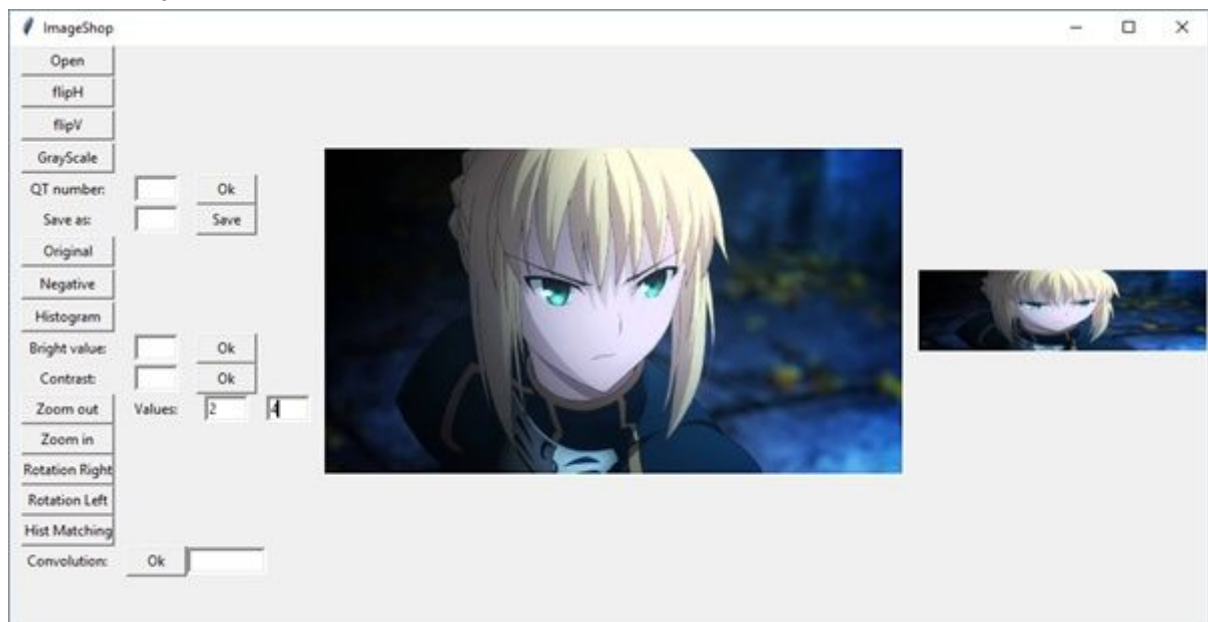
Parte 2

7) Reduzir uma imagem (zoom out) utilizando fatores de redução s_x e s_y (s_x e s_y 1), onde s_x não é necessariamente igual a s_y . Para tanto, defina um retângulo com dimensões s_y e s_x , e mova-o sobre a imagem, de modo que os retângulos nunca se sobreponham e que nenhum pixel da imagem deixe de ser coberto. Para cada posição do retângulo, calcule as médias (R, G e B) dos pixels sob o retângulo, utilizando estes resultados na imagem de saída. Caso o retângulo vá além das dimensões da imagem original, calcule as médias usando apenas os pixels sob o retângulo (25 pontos)

Para s_x e s_y iguais:

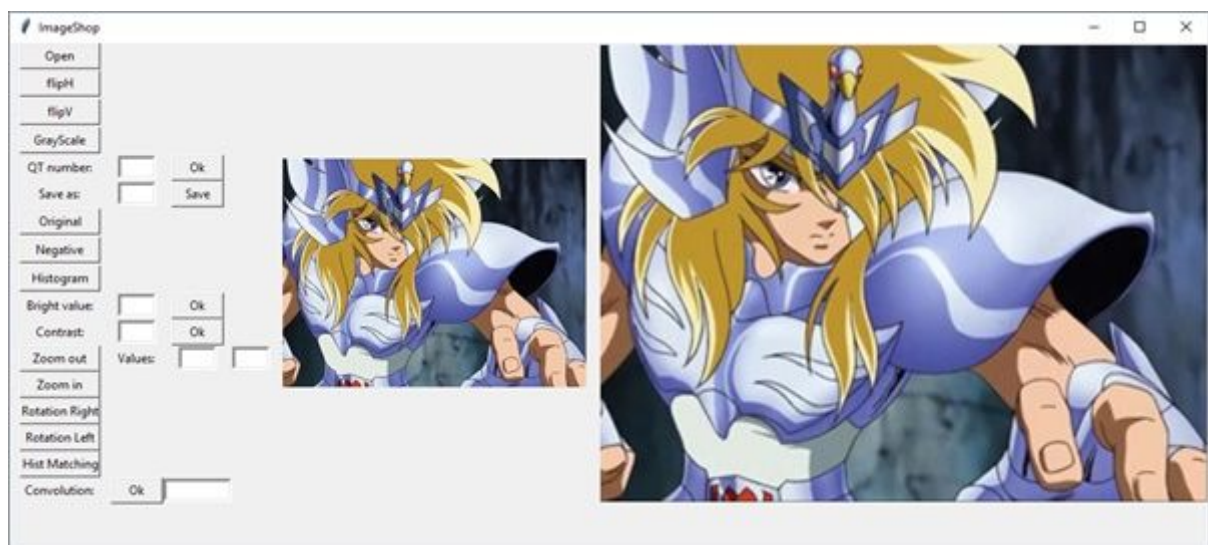


Para s_x e s_y diferentes:

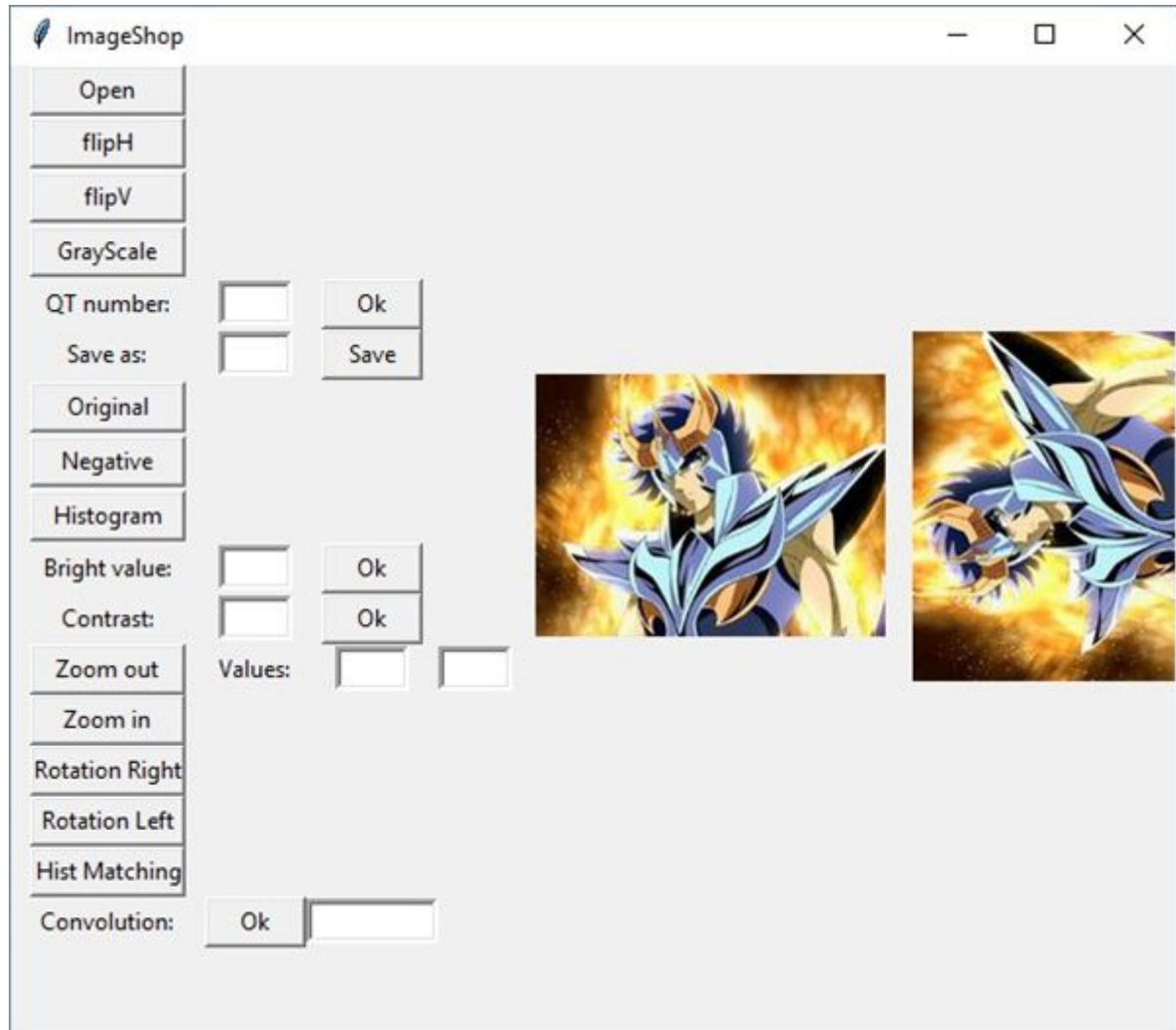


8) Ampliar a imagem (zoom in) utilizando um fator de 2x2 a cada vez. Visto que a imagem será 4 vezes maior que a original, certifique-se de que você alocará memória para tal, apropriadamente. Implemente o processo de ampliação utilizando operações 1-D em dois passos. Primeiro, insira uma linha e uma coluna em branco entre cada duas linhas e colunas originais, respectivamente (veja ilustração na figura a seguir). Após, linearmente interpole os valores para preencher os espaços ao longo de todas as linhas (ou colunas) e, finalmente, ao longo de todas as colunas (ou linhas) (25 pontos).

O fator 2x2 pode ser aplicado diversas vezes.

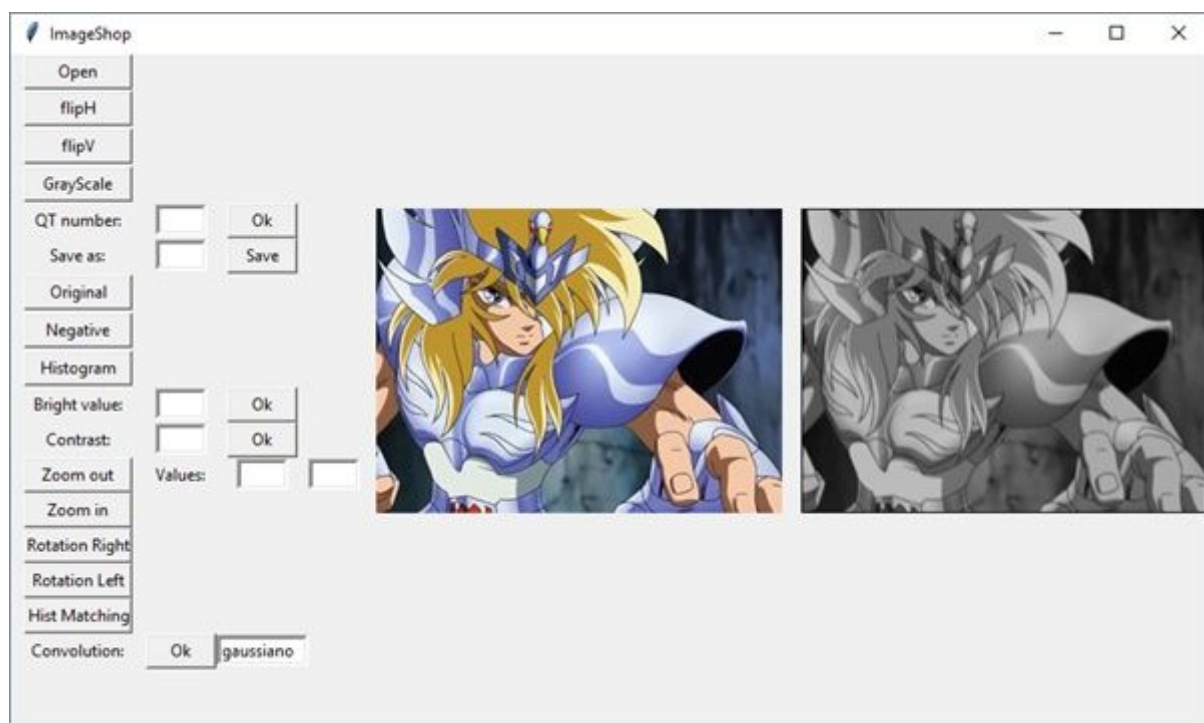


9) Implemente rotação de imagem de 90o (tanto no sentido horário como no sentido anti-horário). A operação de rotação deve poder ser aplicada múltiplas vezes de modo a permitir obter rotações de +/- 180o , +/- 270o , etc. (15 pontos)
Apesar a imagem não ser limpa na rotação, a rotação foi implementada com sucesso.

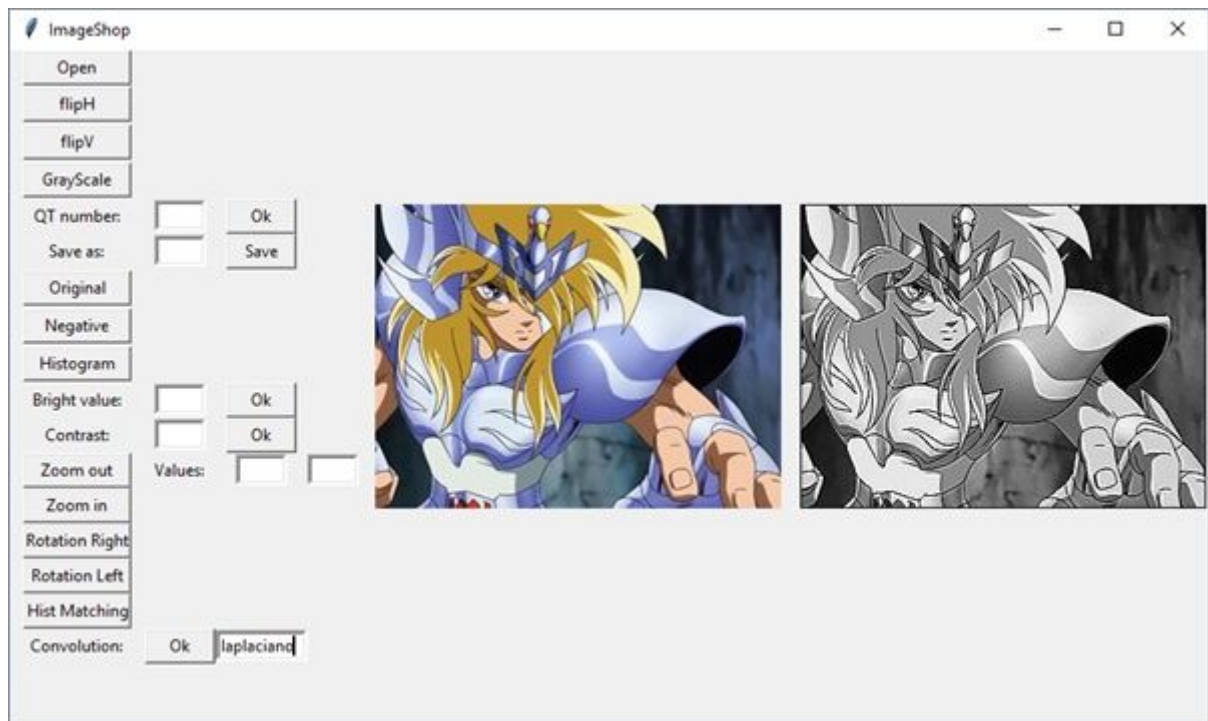


10) Implemente um procedimento para realizar convolução entre uma imagem e um filtro 3x3 arbitrário. De modo a simplificar sua implementação, ignore as bordas da imagem original e aplique a convolução apenas no seu interior. Além disso, aplique a convolução apenas a imagens de luminância. A figura a seguir ilustra a aplicação de uma operação de convolução ao pixel E utilizando um kernel 3x3. Note que o kernel é rotacionado de 180º antes da aplicação. No exemplo do pixel E, o resultado é: $\text{Conv}(E) = iA + hB + gC + fD + eE + dF + cG + bH + aI$. Devido à soma de vários produtos e à possibilidade de utilização de pesos negativos, o resultado da convolução pode ser maior que 255 ou menor que zero. Na sua implementação, você deverá aproximar estes casos com os valores 255 e zero, respectivamente, para os casos (i) a (iii) abaixo. Para os demais, some 127 ao resultado da convolução antes de fazer esta aproximação (clamping). A interface do seu programa deverá aceitar valores arbitrários para os pesos dos filtros. Teste o seu procedimento com os kernels (i) a (vii) a seguir (35 pontos)

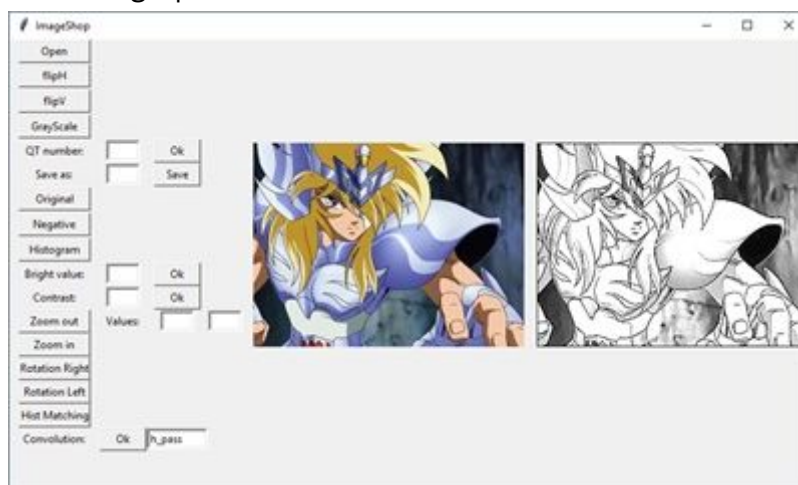
Convolução com kernel gaussiano



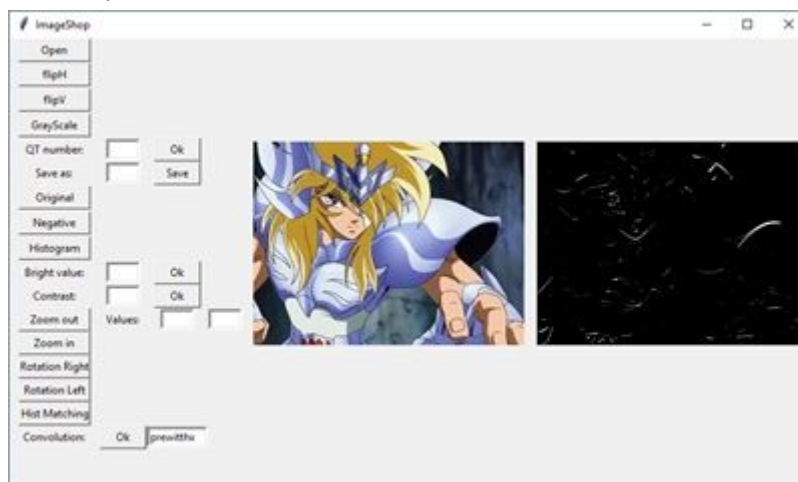
kernel laplaciano:



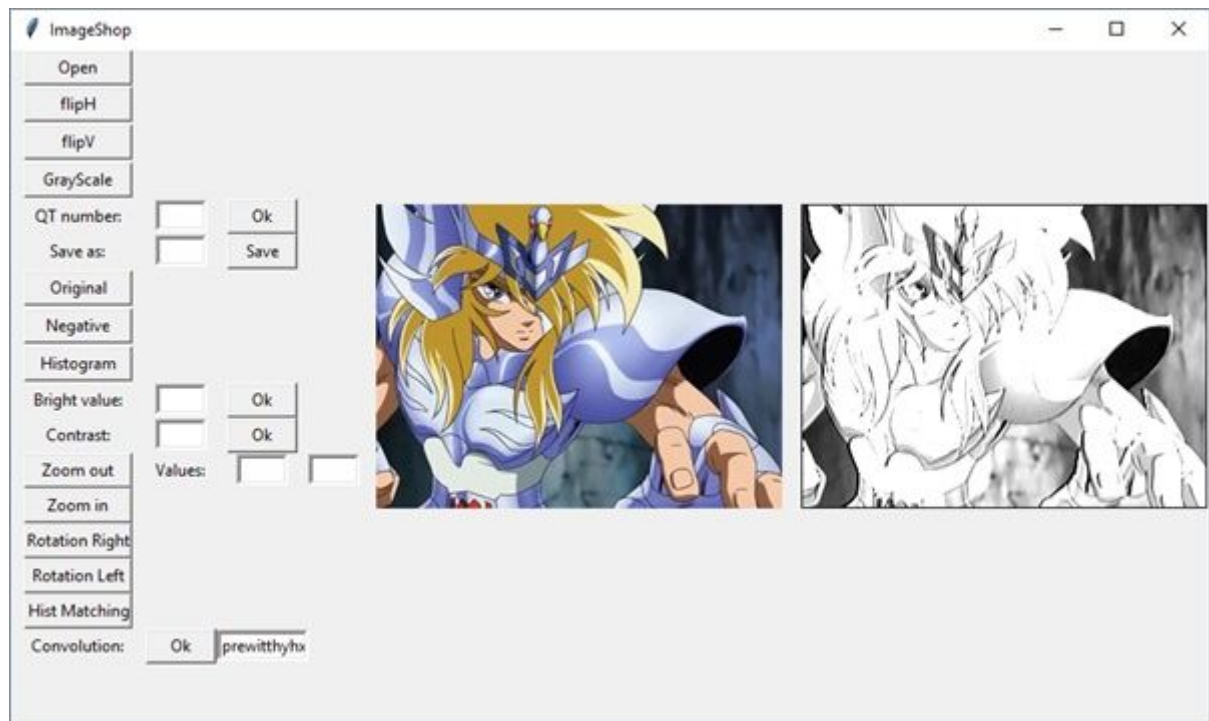
kernel high pass:



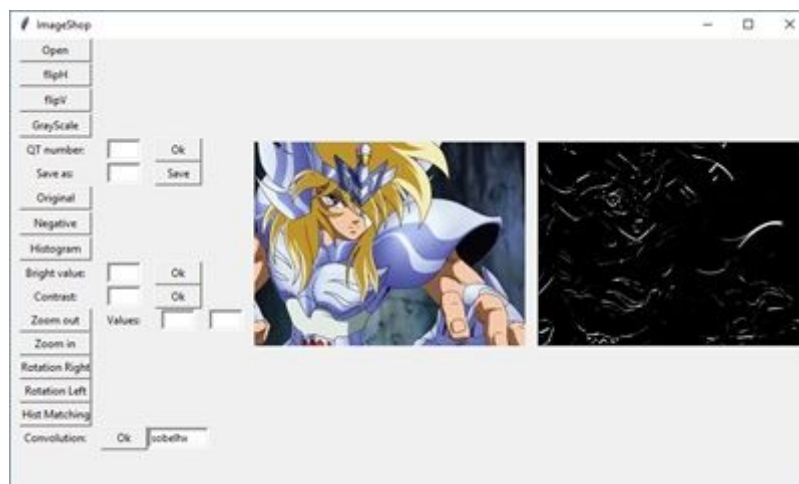
kernel prewitt hx:



Convolução com kernel Prewitthyhx



kernel sobel hx:



kernel sobel hy:



