

Parte I - Completo Satisfatoriamente

1) Calcular e exibir o histograma de uma imagem em tons de cinza - Completo Satisfatoriamente

Foi utilizado para plotar o gráfico o módulo charts da biblioteca QT, atualmente muitos módulos foram separados da instalação principal para torná-la mais leve, e precisam ser instalados manualmente.

Dessa forma a operação GrayScale foi ajustada para retornar agora, o histograma da imagem calculado durante a conversão, e os valores mínimo e máximo para operação de quantização. O programa então, gera através de uma versão em escala de cinza da imagem de origem o seu histograma em uma nova janela utilizando um chart, outras estruturas para título, eixos e própria janela, do tipo dialog. A altura das colunas foi normalizada para 1, utilizando o tom mais frequente na imagem para esta normalização, tendo então variações de 0 a 1.

OBS: A ação de visualizar o histograma preserva a última operação realizada na imagem, logo ela utiliza uma imagem em escala de cinza, mas não aplica a operação na cópia que estamos trabalhando.

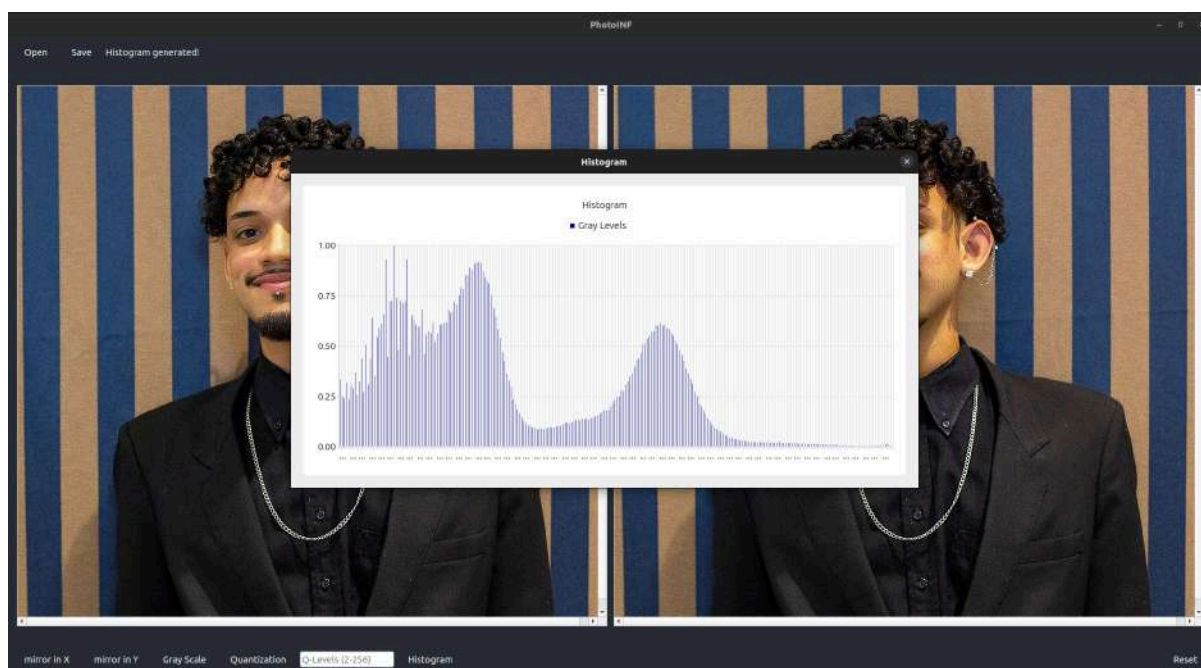


Ilustração 1: Visualização do histograma de escalas em cinza de uma imagem colorida gerada pelo programa

2) Ajustar o brilho de uma imagem - **Completo Satisfatoriamente**

Implementado a operação de ajuste de brilho somando um valor entre -255 a 255 a todos os pixels da imagem, utilizando uma função clamp para definir os máximos e mínimos entre 0 e 255.

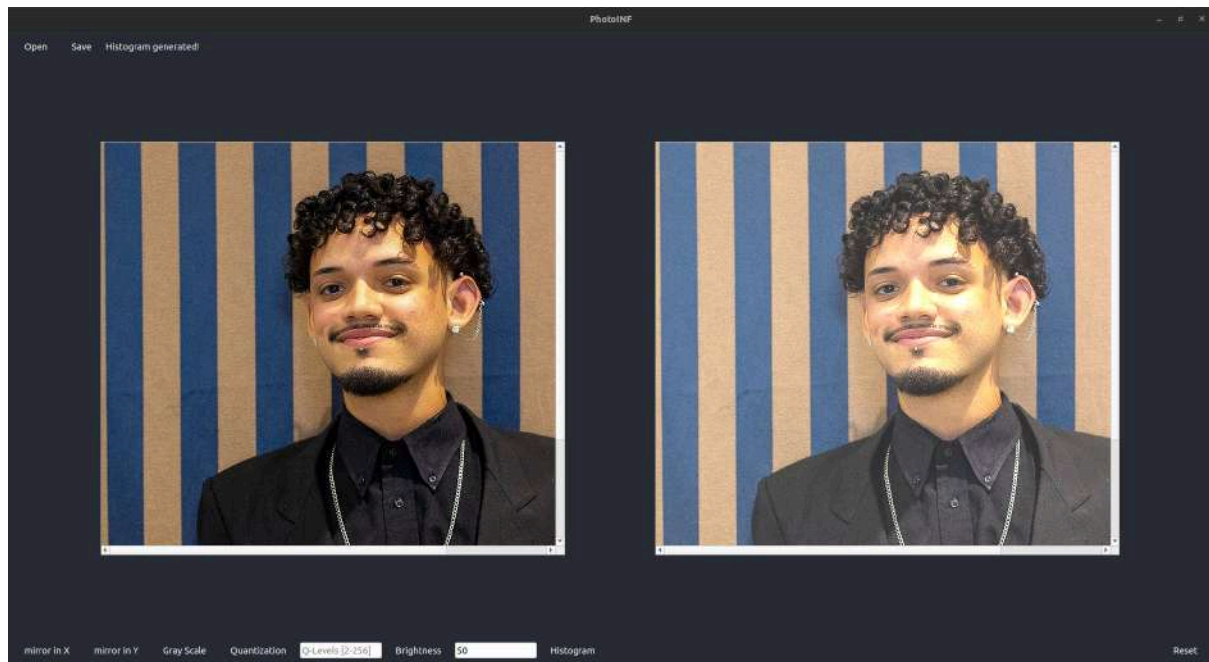


Ilustração 2: Brilho da imagem ajustado em +50.

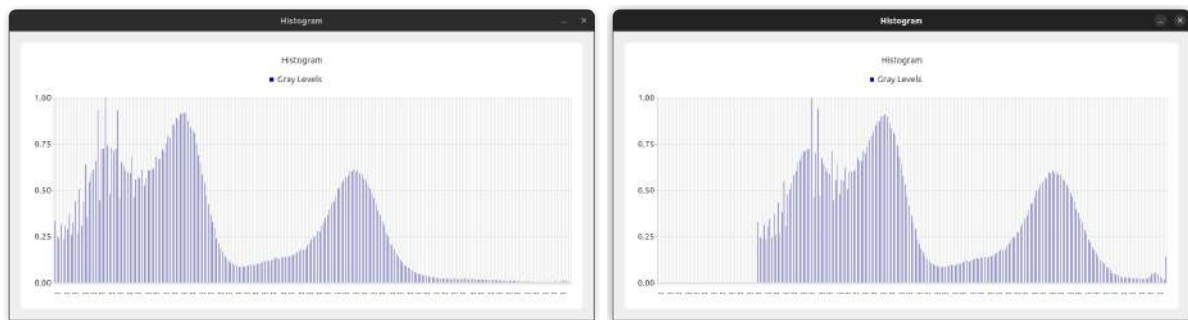


Ilustração 3: comparação do histograma gerado para imagem original e com ajuste de brilho em +50.

3) Ajustar o brilho de uma imagem - **Completo Satisfatoriamente**

Implementado a operação de ajuste de contraste multiplicando por um valor entre 0 a 255 todos os pixels da imagem, utilizando uma função clamp para definir os máximos e mínimos entre 0 e 255.

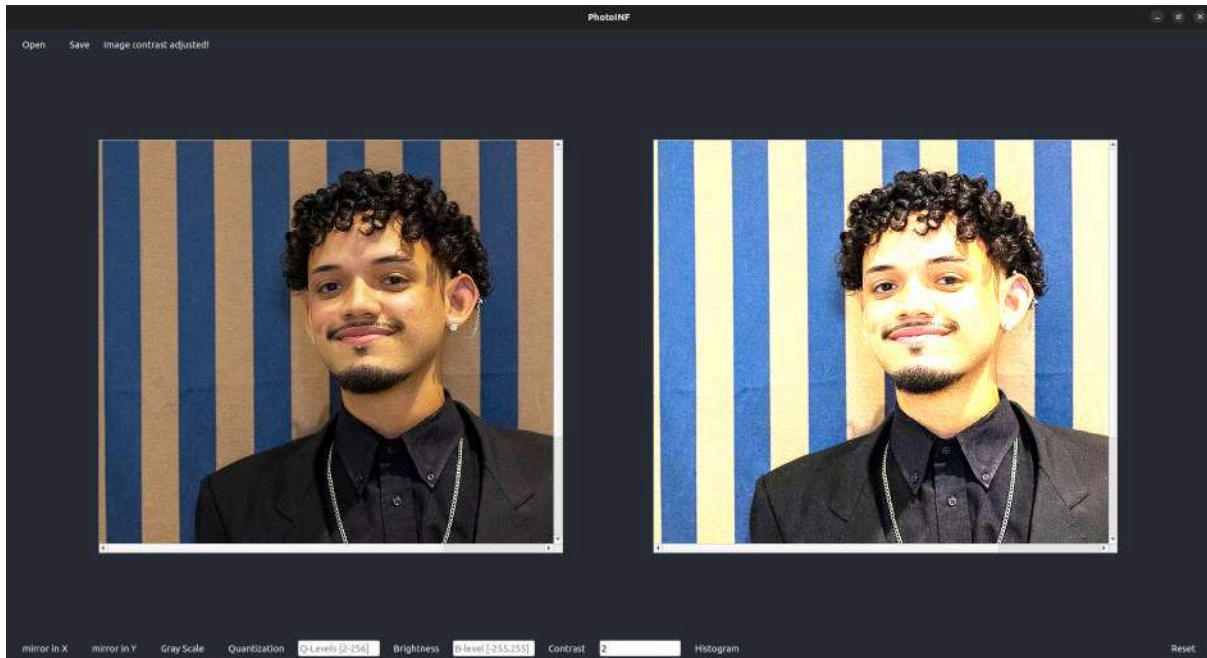


Ilustração 4: Contraste da imagem ajustado por 2.

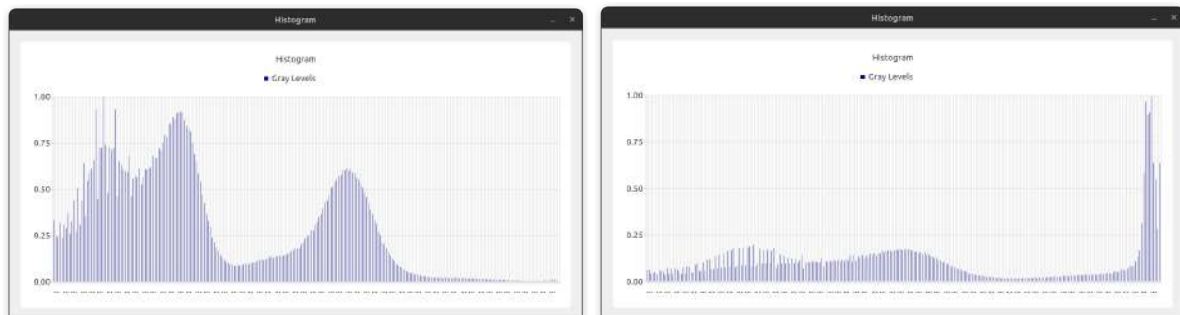


Ilustração 5: comparação do histograma gerado para imagem original e com ajuste de contraste por 2.

4) Calcular e exibir o negativo de uma imagem - **Completo Satisfatoriamente**

Implementado a operação de negativo, todos os pixels têm suas tonalidades subtraídas de 255.

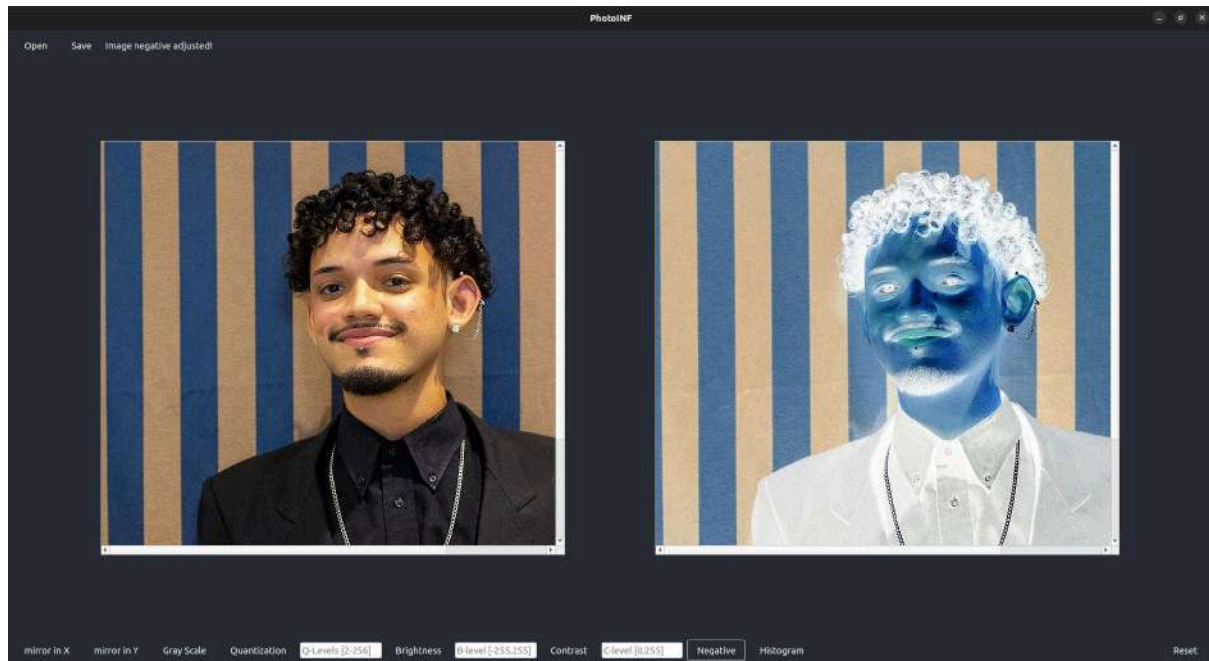


Ilustração 6: Cálculo do negativo da imagem.

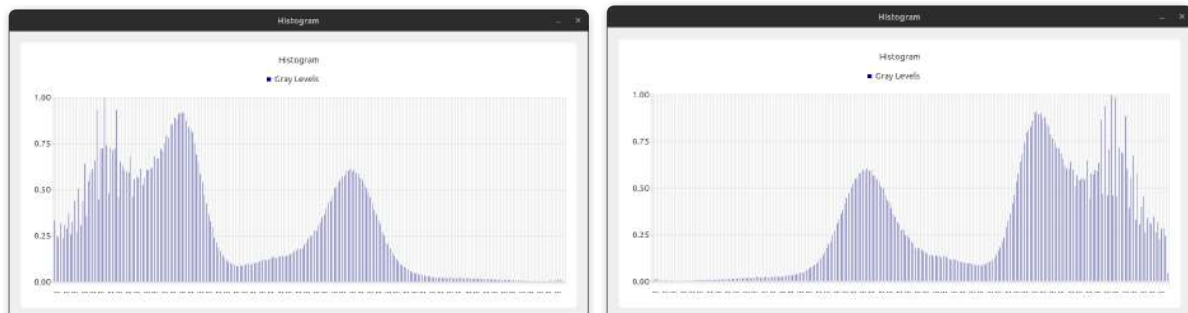


Ilustração 5: comparação do histograma gerado para imagem original e negativa e evidência da sua relação de inversão.

5) Equalizar o histograma de uma imagem - **Completo Satisfatoriamente**

Implementado a operação de equalização. A partir do código apresentado em aula foi implementado o algoritmo, tendo como ponto de partida que as fotos ainda que em gray scale possuem 3 componentes R G e B, logo precisamos da componente do histograma cumulativo corresponde ao tom de cinza da combinação destas cores em cinza.

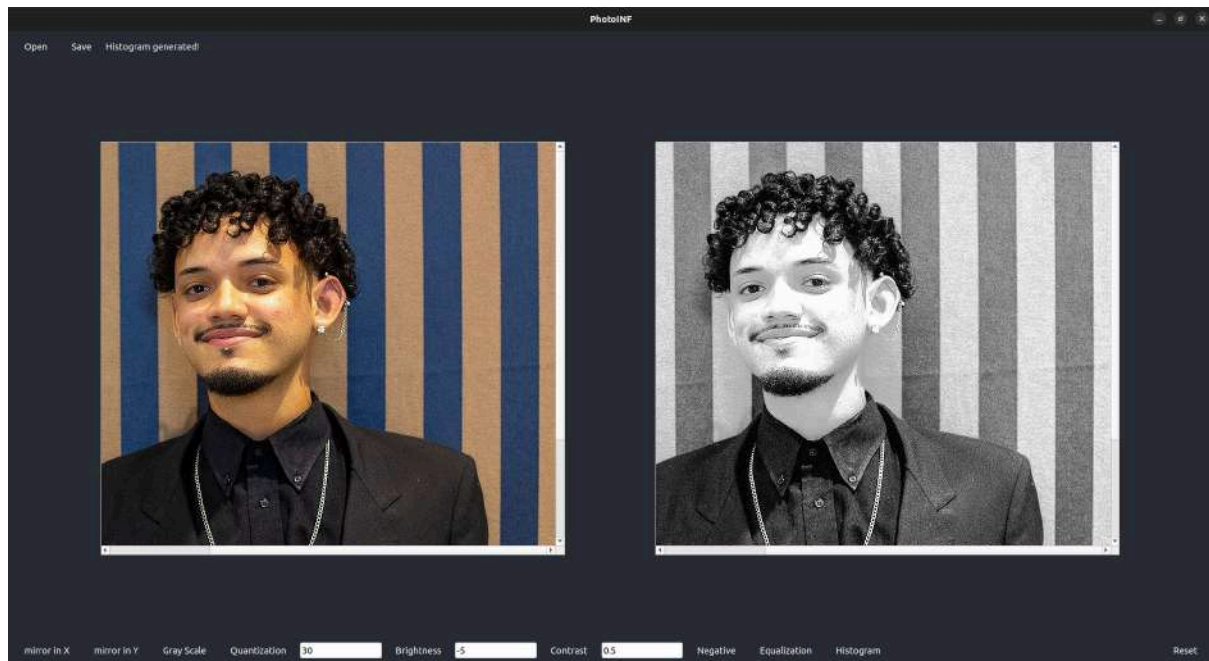


Ilustração 8: Imagem equalizada em tons de cinza

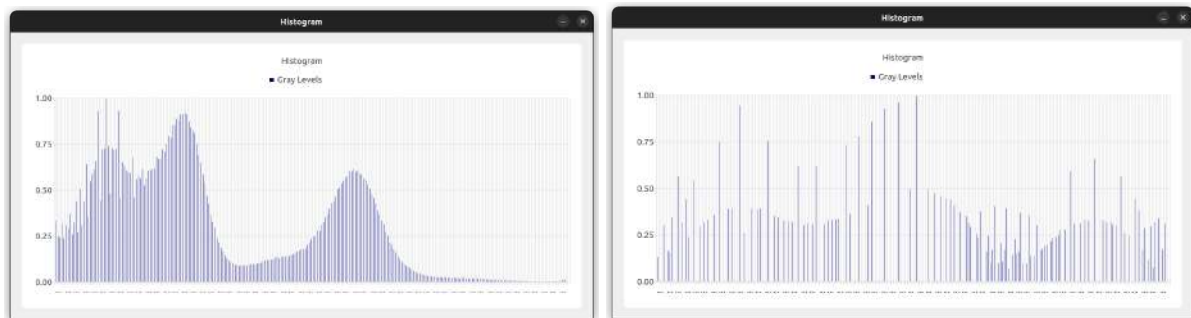


Ilustração 9: comparação entre o histograma da imagem original e o equalizado (utilizando o histograma cumulativo).

6) Realizar Histogram Matching de pares de imagens em tons de cinza - Completo Satisfatoriamente

Implementado a operação de histogram matching. O programa exibe uma view escalada da imagem alvo em miniatura entre a imagem original e a cópia para sugerir uma operação de matching entre elas. O código dispõe de duas estruturas QImage para preservar a imagem target original, e assim gerar seu histograma. As operações da imagem target ficam no canto superior direito da janela. Além disso foram feitas alterações nas funções grayscale e histogram, para que elas suportem receber por parâmetro as imagens que irão manipular e assim poder operar sobre diferentes imagens, uma vez que agora quero gerar o histograma de duas imagens diferentes.

O algoritmo foi implementado como sugere nos slides de aula, são declarados e inicializados os histogramas da imagem original e da imagem alvo, calculados seus histogramas cumulativos normalizados, para o caso de diferentes resoluções. Em seguida, calcula-se para cada valor do histograma cumulativo normalizado da imagem original o valor mais próximo no histograma cumulativo normalizado da imagem alvo. Por fim, é feita a atribuição a cada pixel os tons combinados.

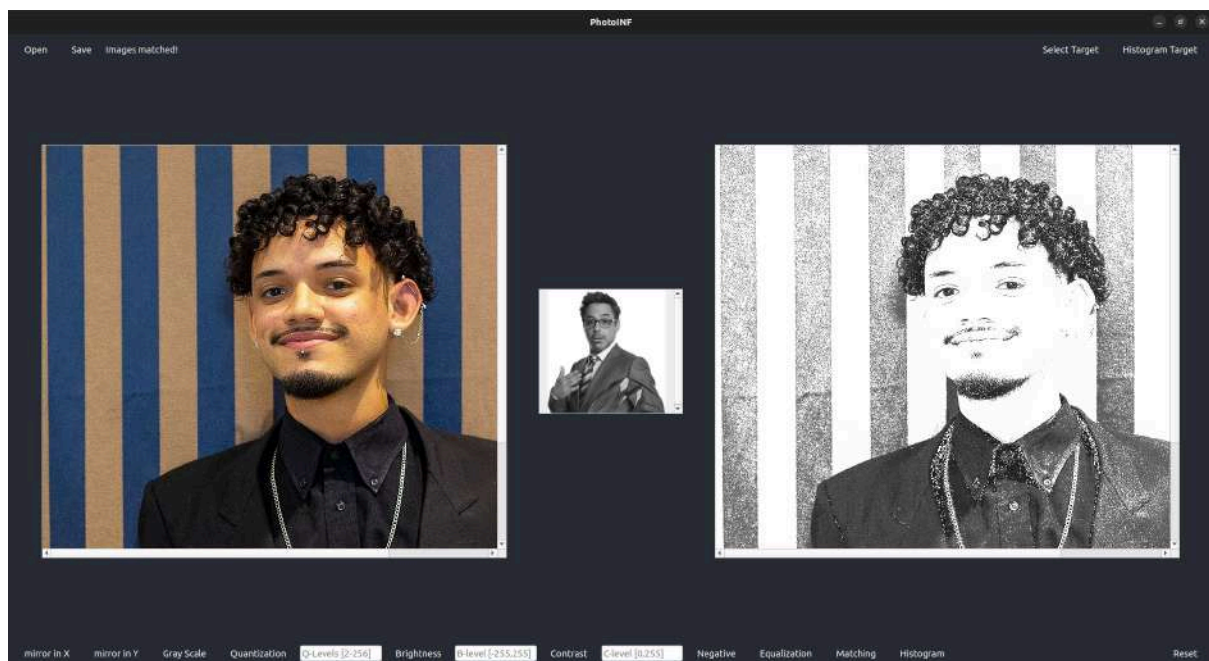


Ilustração 10: Matching de uma imagem target localizada no centro com a imagem principal à esquerda.

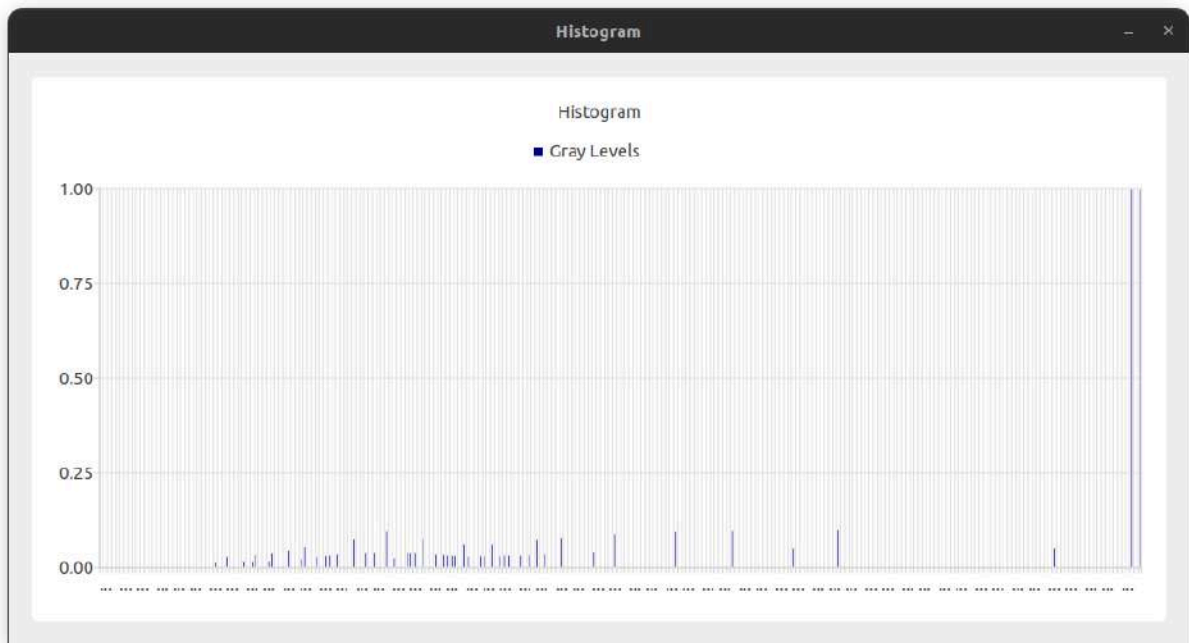


Ilustração 11: Histograma da imagem principal após o matching (o histograma da imagem original antes da operação pode ser visualizado em ilustrações anteriores).

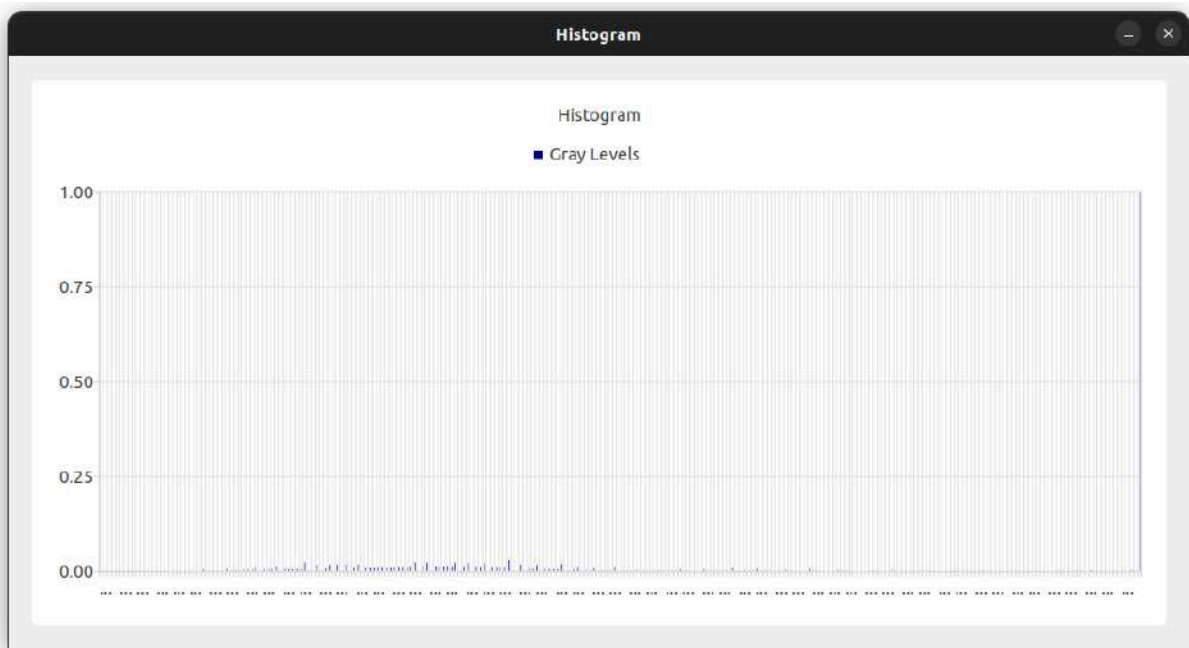


Ilustração 11: Histograma da imagem alvo (Robert Downey Jr.).

Parte II - Completo Satisfatoriamente

7) Reduzir uma imagem (zoom out) utilizando fatores de redução s_x e s_y - Completo Satisfatoriamente

Implementado a operação de zoom out. O programa cria uma nova imagem de largura e altura reduzidos, respectivamente, em fatores s_x e s_y , a partir daí ele começa a percorrer a imagem fonte em blocos s_x por s_y , enquanto percorre os pixels da imagem de destino. A cada bloco é calculado a média R, G e B que se tornam os componentes RGB do respectivo pixel na imagem de destino. A maior dificuldade é entender a maneira com que os blocos caminham pela imagem original ao mesmo tempo em que estamos percorrendo a imagem de destino.

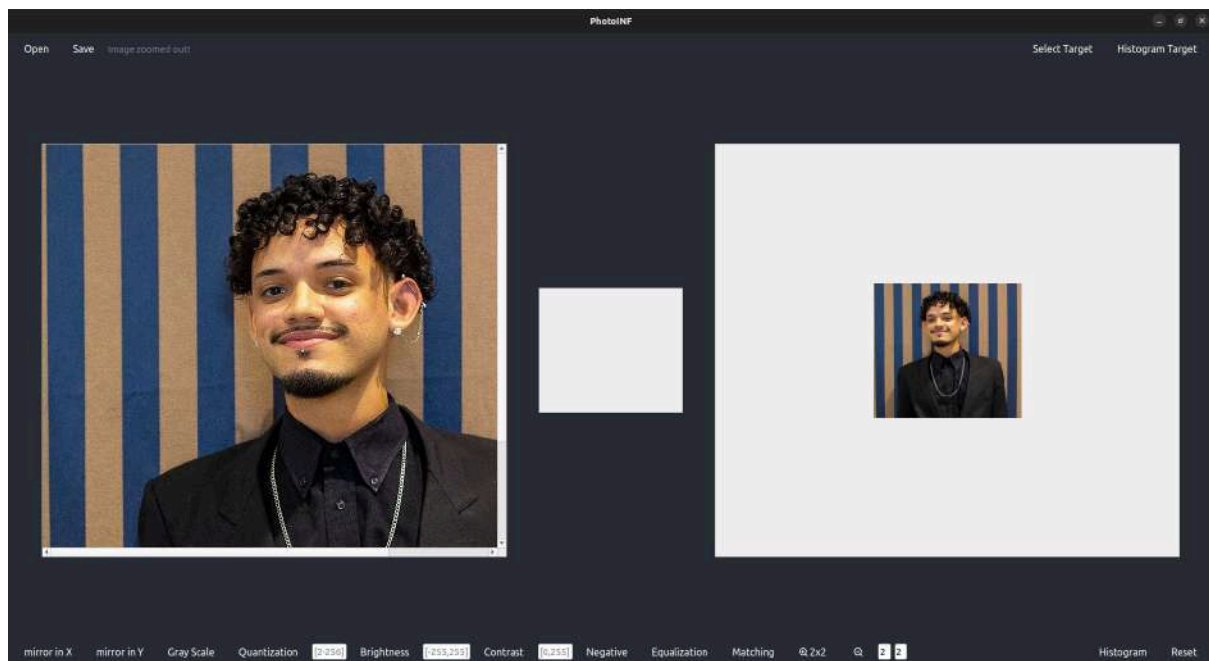


Ilustração 12: Imagem com zoom out de 2 por 2.

8) Ampliar a imagem (zoom in) utilizando um fator de 2x2 a cada vez - Completo Satisfatoriamente

Implementado a operação de zoom in. O programa cria uma nova imagem de largura e altura aumentados em, respectivamente, largura-1 e altura-1. Dessa forma temos uma imagem grande o suficiente para alternar linhas entre as colunas e linhas da imagem original. Após isso, é recuperado as colunas e linhas da imagem fonte de forma intercalada na imagem destino. Em seguida, as colunas ímpares são interpoladas com as colunas pares nas linhas pares. Por fim, as linhas ímpares são todas interpoladas com as linhas ímpares, gerando uma imagem aumentada através da interpolação de pixels.

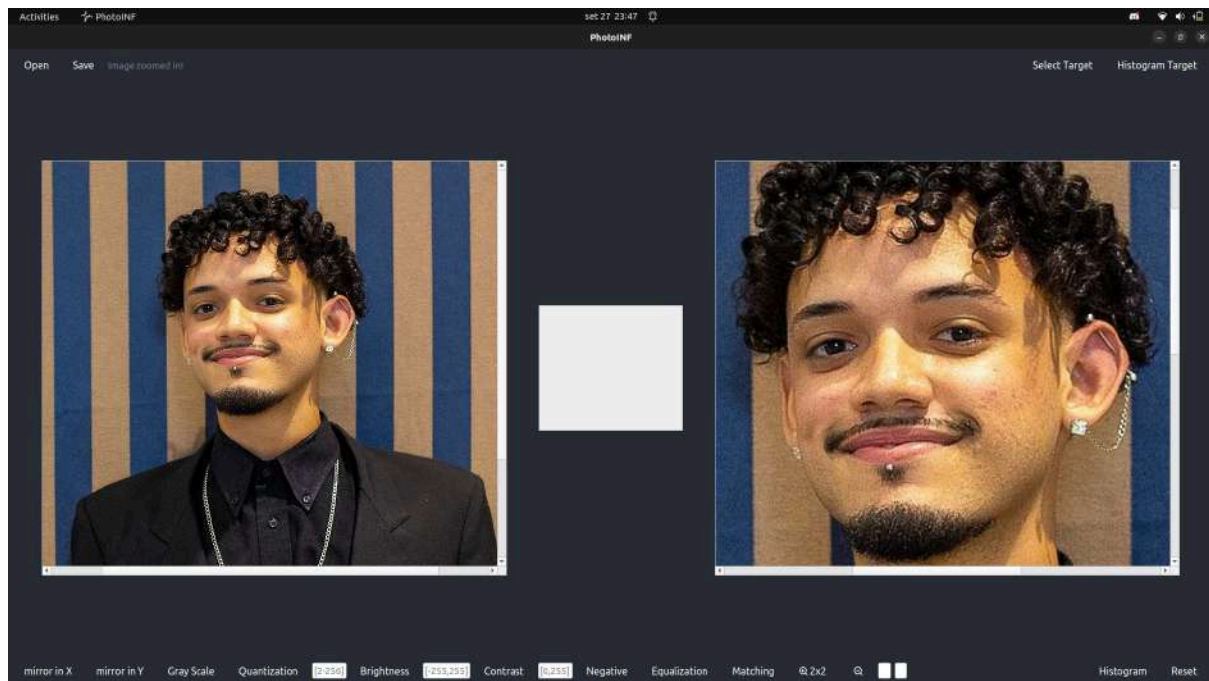


Ilustração 13: Imagem com zoom in de 2 por 2

9) Rotacionar imagem de 90ª (tanto no sentido horário como no sentido anti horário) - **Completo Satisfatoriamente**

Implementado a operação de rotação em ambos os sentidos. O programa cria uma nova imagem invertendo largura e altura da imagem original, após isso, para cada pixel i,j na imagem original calcula a rotação para a imagem destino.

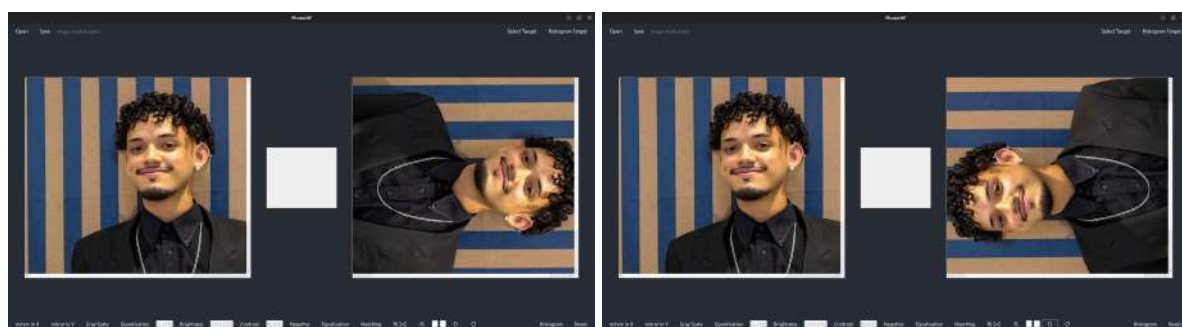


Ilustração 14: Imagem rotacionada em 90º e -90º.

9) Implementar um procedimento para realizar convolução entre uma imagem e um filtro 3x3 arbitrário - **Completo Satisfatoriamente**

Aqui foi implementada a operação de convolução. Utilizando botões auxiliares de preenchimento é possível selecionar um kernel pré estabelecido. A operação é implementada como indica o algoritmo visto em aula, cada pixel tem seu tom calculado a partir do somatório das multiplicações feitas a partir da sobreposição do kernel sobre aquele pixel e seus vizinhos. A parte mais complicada foi entender a maneira correta de ajustar e realizar o clamping dos resultados que utilizam filtros baseados em passa altas.

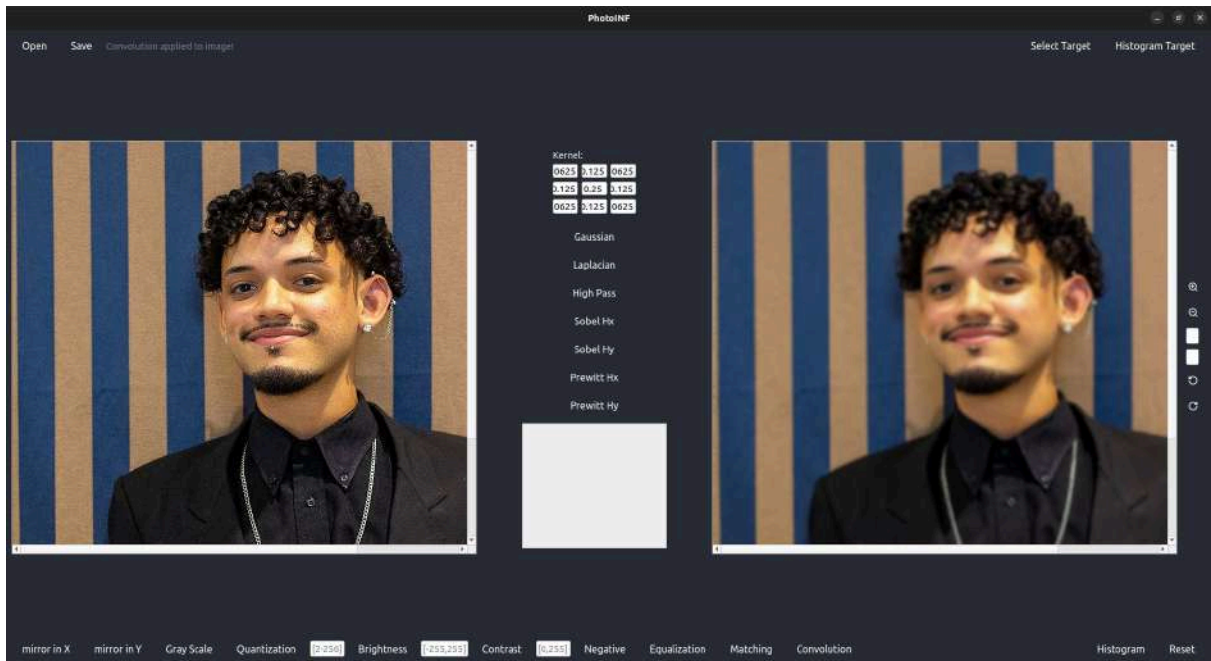


Ilustração 15: Imagem convolucionada com um filtro gaussiano 10 vezes.



Ilustração 16: resultado de imagens convolucionadas com os filtros passa-altas genérico, laplacian, prewitt hy e sobel hx, respectivamente.