Conversor de imagem

Nomes:	RGMs:
Eduarda Fernandes	29204356
Davi Santos de Andrade	31075550
Johnatan Caetano	30087155
Everman	30333717
Daniel Medeiros	29381169

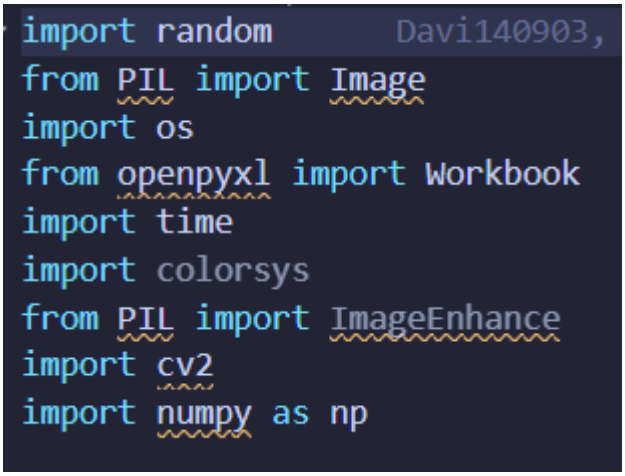
- Objetivo:
  - Aplicação de uma função para aumentar o contraste no processamento da imagem nas escalas de cinza.
  - Aplicação de uma função para aumentar o contraste no processamento da imagem colorida.

- Equipamentos/ferramentas utilizados:

  Para desenvolver este conversor conforme descrito anteriormente, utilizamos um notebook Samsung Core i5 de décima geração como plataforma principal. Optamos pelo Visual Studio Code devido à sua interface intuitiva e fácil manipulação. A linguagem de programação escolhida foi o Python, devido à sua versatilidade e eficiência. Para garantir uma organização adequada do projeto, decidimos utilizar o GitHub para armazenar e colaborar no código, permitindo que todos os membros do grupo pudessem contribuir de forma eficiente.

- Procedimento experimental:

  ➔ Para este projeto, utilizamos a linguagem Python e começamos a fazer as importações das bibliotecas necessárias para que o código seja executado corretamente.



➔ Na segunda parte do código, desenvolvemos uma função que carrega uma imagem a partir do arquivo especificado.

Esta função é crucial para realizar a busca da imagem no diretório onde o projeto está sendo executado. Se a imagem for encontrada, ela será retornada; caso contrário, uma mensagem de erro será gerada.

```
def carregar_imagem(caminho):
    if not os.path.exists(caminho):
        raise FileNotFoundError(f"Arquivo '{caminho}' não encontrado.")

    try:
        imagem = Image.open(caminho)
        return imagem
    except Exception as e:
        print("Ocorreu um erro ao carregar a imagem:", e)

def imagem_para_matriz(imagem):
    largura, altura = imagem.size
    matriz = []

    for y in range(altura):
        linha = []
        for x in range(largura):
            pixel = imagem.getpixel((x, y))
            linha.append(pixel)
        matriz.append(linha)

    return matriz
```

No terceiro trecho do código, elaboramos uma função para pegar as dimensões, altura e largura, da imagem selecionada. Em seguida, declaramos uma matriz cujas dimensões são definidas com base no tamanho da imagem.

```
def imagem_para_matriz(imagem):
    largura, altura = imagem.size
    matriz = []

    for y in range(altura):
        linha = []
        for x in range(largura):
            pixel = imagem.getpixel((x, y))
            linha.append(pixel)
        matriz.append(linha)

    return matriz
```

No quarto trecho do código, elaboramos uma função para verificar a cor de cada pixel da imagem para no próximo passo conseguirmos mexer no contraste tanto da escala de cinza, quanto da imagem colorida.

```
def verificar_cor_pixel(imagem, x, y):
    pixel = imagem.getpixel((x, y))
    print("Pixel selecionado:")
    print(f"A cor do pixel na posição ({x}, {y}) é: {pixel}")
    time.sleep(1)

    largura, altura = imagem.size

    pixels_mesma_cor = []
    for py in range(altura):
        for px in range(largura):
            if imagem.getpixel((px, py)) == pixel:
                pixels_mesma_cor.append((px, py))

    print("Número de pixels com a mesma cor: Selecionados!")
    time.sleep(1)

    return pixel, pixels_mesma_cor

def alterar_cor_pixels(imagem, pixels, nova_cor):
    for x, y in pixels:
        imagem.putpixel((x, y), nova_cor)
```

No quinto trecho do código, elaboramos uma função para pegar a cor de cada pixel da imagem e aumentar o contraste da imagem em escala de cinza, realizando o cálculo do histograma e do CDF para aí sim realizar o cálculo da imagem final.

```python
#FUNÇÃO PARA AUMENTAR CONTRASTE DE IMAGEM EM ESCALA DE CINZA
def aumentar_contraste_escala_cinza(imagem):
    largura, altura = imagem.size

    if imagem.mode != 'L':
        imagem = imagem.convert('L')

    #CALCULO HISTOGRAMA
    histograma = [0]*256
    for y in range(altura):
        for x in range(largura):
            intensidade = imagem.getpixel((x,y))
            histograma[intensidade] += 1

    total_pixels = altura*largura
    frequencias_normalizadas = [freq/total_pixels for freq in histograma]

    #CALCULO CDF
    cdf = [sum(frequencias_normalizadas[:i+1]) for i in range(256)]

    #CALCULO IMAGEM FINAL
    for y in range(altura):
        for x in range(largura):
            intensidade = imagem.getpixel((x,y))
            novo_valor = int(cdf[intensidade]*255)
            imagem.putpixel((x,y), novo_valor)

    return imagem
```

Logo depois, criamos a funcionalidade que faz o cálculo do aumento de contraste da imagem colorida, e retorna a mesma com a alteração concluída.

```python
#FUNÇÃO PARA AUMENTAR CONTRASTE DE IMAGEM COLORIDA
def aumentar_contraste_colorido(imagem, alpha=1.0, beta=0.0):
    # Aplicando a transformação de contraste
    nova_imagem = np.clip(alpha * imagem + beta, 0, 255).astype(np.uint8)
    return nova_imagem
```

No método principal do programa, criamos a integração com a nova parte solicitada, com a interação do usuário para a execução e a chamada das funções criadas anteriormente para o cumprimento do objetivo.

```
#PARTE 4 (AUMENTAR CONTRASTE EM ESCALA CINZA)
opcao_parte4 = input("Deseja executar a Parte 4 (Aumentar o contraste da imagem em escala cinza)? (s/n): ").lower()
if opcao_parte4 == 's':
    print("PARTE 4: AUMENTAR O CONTRASTE DAS IMAGENS EM ESCALA DE CINZA")
    nome_arquivo = input("Digite o nome do arquivo de imagem (sem o formato): ")
    caminho_png = nome_arquivo + ".png"
    caminho_jpg = nome_arquivo + ".jpg"
    if os.path.exists(caminho_png):
        caminho_imagem = caminho_png
    elif os.path.exists(caminho_jpg):
        caminho_imagem = caminho_jpg
    else:
        print("Arquivo não encontrado.")
        return
    imagem = carregar_imagem(caminho_imagem)
    if not imagem:
        return
    try:
        fator_contraste = float(input("Digite o fator de contraste desejado (por exemplo, 1.5): "))
        if fator_contraste <= 0:
            raise ValueError("O fator de contraste deve ser maior que zero.")


        imagem = aumentar_contraste_escala_cinza(imagem)
        nome_arquivo_contraste_escala_cinza = nome_arquivo + "_contraste_aumentado.png"
        imagem.save(nome_arquivo_contraste_escala_cinza)
        print(f"Imagem com contraste aumentado salva como: {nome_arquivo_contraste_escala_cinza}")
    except ValueError as ve:
        print("Erro", ve)
elif opcao_parte4 == 'n':
    print("Você optou por pular a Parte 4.")
else:
    print("Opção inválida")
```

```
# PARTE 5 (AUMENTAR O CONTRASTE DA IMAGEM COLORIDA)
opcao_parte5 = input("Deseja executar a Parte 5 (Aumentar o contraste da imagem colorida)? (s/n): ").lower()
if opcao_parte5 == 's':
    print("PARTE 5: AUMENTAR O CONTRASTE DAS IMAGENS COLORIDAS")
    nome_arquivo = input("Digite o nome do arquivo de imagem (sem o formato): ")
    caminho_png = nome_arquivo + ".png"
    caminho_jpg = nome_arquivo + ".jpg"
    if os.path.exists(caminho_png):
        caminho_imagem = caminho_png
    elif os.path.exists(caminho_jpg):
        caminho_imagem = caminho_jpg
    else:
        print("Arquivo não encontrado.")
        exit()
    def carregar_imagem(caminho):
        imagem = cv2.imread(caminho)
        if imagem is None:
            print("Erro ao carregar a imagem.")
            return None
        else:
            return imagem
    imagem = carregar_imagem(caminho_imagem)
if imagem is not None:
    try:
        imagem = aumentar_contraste_colorido(imagem, alpha=1.5, beta=10)
        nome_arquivo_contraste_rgb = nome_arquivo + "_contraste_aumentado_colorido.png"
        cv2.imwrite(nome_arquivo_contraste_rgb, imagem)
        print(f"Imagem com contraste aumentado salva como: {nome_arquivo_contraste_rgb}")
    except Exception as e:
        print("Erro:", e)
elif opcao_parte5 == 'n':
    print("Você optou por pular a Parte 5.")
else:
    print("Opção inválida")
```

Analisando os resultados, percebe-se que o código funcionou de maneira correta, realizando o aumento de contraste tanto nas imagens em escala de cinza, quanto nas imagens coloridas.

- Considerações finais:

  Uma nova etapa foi integrada ao projeto. Agora, conseguimos aumentar o contraste em imagens com escala de cinza e imagens coloridas. É uma etapa interessante, pois sempre foi possível a realização da edição de fotos e imagens nos apps e programas, seja no celular ou no computador, e a partir deste trabalho foi possível entender como funciona o processo de contraste, e o que acontece com a imagem quando se aumenta o mesmo.