

Conversor de imagem

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

- **Objetivo:**

- Transformar uma imagem do sistema RGB para o sistema CMYK;
- Passa a imagem para diferentes tipos de escalas de cinza:
 - Média ponderada;
 - Luminosidade ;
 - Dessaturação e
 - Decomposição de cores(máximo e mínimo)

- **Equipamentos/ferramentas utilizados:**

Para desenvolver este conversor de imagem em diferentes escalas de cinza e para o sistema CMYK, 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.

```
Davi140903, ontem | Python (Davi140903)
import random          Davi140903, ontem • conversor
from PIL import Image
import os
from openpyxl import Workbook
import time
import colorsys
```

→ 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 converter para o CMYK.

```
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 converter de RGB para CMYK.

```
#CONVERSOR DE RGB PARA CMYK
def rgb_to_cmyk(rgb):
    r, g, b = rgb
    c = 1 - (r / 255)
    m = 1 - (g / 255)
    y = 1 - (b / 255)
    k = min(c, m, y)
    if k == 1:
        return 0, 0, 0, 1
    return (c - k) / (1 - k), (m - k) / (1 - k), (y - k) / (1 - k), k
```

Logo depois, criamos as funcionalidades que convertem de RGB para as escalas de cinzas, separadas pelos comentários nos códigos.

```
#FUNÇÕES DE CONVERSÃO DE RGB PARA ESCALAS DE CINZA
def rgb_para_escalade_cinza_media_ponderada(rgb):
    r, g, b = rgb
    y = 0.299 * r + 0.587 * g + 0.114 * b
    return int(y), int(y), int(y)

def rgb_para_escalade_cinza_luminosidade(rgb):
    r, g, b = rgb
    y = 0.21 * r + 0.72 * g + 0.07 * b
    return int(y), int(y), int(y)

def rgb_para_escalade_cinza_dessaturacao(rgb):
    r, g, b = rgb
    y = (r + g + b) / 3
    return int(y), int(y), int(y)

def rgb_para_escalade_cinza_maximo(rgb):
    max_value = max(rgb)
    return max_value, max_value, max_value

def rgb_para_escalade_cinza_minimo(rgb):
    min_value = min(rgb)
    return min_value, min_value, min_value

def converter_para_escalade_cinza(imagem, metodo):
    largura, altura = imagem.size
    for y in range(altura):
        for x in range(largura):
            pixel = imagem.getpixel((x, y))
            nova_cor = metodo(pixel)
            imagem.putpixel((x, y), nova_cor)
```

E a função que usa cada método de conversão de cinza e converte todos de uma vez só.

```
#FUNÇÃO QUE USA CADA METODO DE CONVERSÃO DE CINZA E CONVERTE TODOS DE UMA VEZ
def converter_todos_para_escalade_cinza(imagem, nome_arquivo):
    # Métodos de conversão de escala de cinza
    metodos = {
        "Média Ponderada": rgb_para_escalade_cinza_media_ponderada,
        "Luminosidade": rgb_para_escalade_cinza_luminosidade,
        "Dessaturação": rgb_para_escalade_cinza_dessaturacao,
        "Decomposição de Cores (Máximo)": rgb_para_escalade_cinza_maximo,
        "Decomposição de Cores (Mínimo)": rgb_para_escalade_cinza_minimo
    }

    for nome_metodo, metodo in metodos.items():
        imagem_temp = imagem.copy()
        converter_para_escalade_cinza(imagem_temp, metodo)
        novo_nome_arquivo = nome_arquivo + f"_convertido_{nome_metodo.replace(' ', '_').lower()}.png"
        imagem_temp.save(novo_nome_arquivo)
        print(f"Imagem convertida para escala de cinza ({nome_metodo}) e salva como: {novo_nome_arquivo}")
```

E por último, mas não menos importante, fizemos a alteração do menu principal, incluindo a terceira etapa do projeto.

```
#PARTE 3 (CONVERSÃO DE RGB PARA CMYK E PARA ESCALAS DE CINZA)
opcao_parte3 = input("Deseja executar a Parte 3 (Converter imagem para CMYK ou escala de cinza)? (s/n): ").lower()
print("PARTE 3: CONVERSÃO DE RGB PARA CMYK OU ESCALA DE CINZA")
time.sleep(1.5)
if opcao_parte3 == 's':
    if not imagem:
        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

    opcao = input("Escolha uma opção:\n1. Converter para CMYK\n2. Converter para Escala de Cinza\nOpção: ")

    if opcao == "1":
        # Código de conversão para CMYK
        img = Image.open(caminho_imagem)
        if img.mode != 'RGB':
            img = img.convert('RGB')
```

```
        cmyk_img = Image.new('CMYK', img.size)
        pixels = cmyk_img.load()

        for i in range(img.size[0]):
            for j in range(img.size[1]):
                rgb = img.getpixel((i, j))
                cmyk = rgb_to_cmyk(rgb)
                pixels[i, j] = tuple(int(255 * x) for x in cmyk)

        cmyk_img.save("output_cmyk.jpg")
        print("imagem convertida com sucesso para CMYK")

    elif opcao == "2":
        # Código de conversão para escala de cinza...
        converter_todos_para_escala_de_cinza(imagem, nome_arquivo)
        return
elif opcao_parte3 == 'n':
    print("Você optou por pular a Parte 3.")
    time.sleep(1.5)
else:
    print("Opção inválida.")

if __name__ == "__main__":
    main()
```

Analisando os resultados, percebe-se que o código funcionou de maneira correta, realizando a conversão das imagens para as escalas de cinza e para o sistema CMYK, conforme o objetivo proposto.

- **Considerações finais:**

Uma nova etapa foi integrada ao projeto. Agora, conseguimos extrair os valores RGB de uma imagem selecionada e convertê-los para o sistema CMYK. Esse processo resulta na alteração da coloração da imagem, ampliando suas possibilidades visuais. Além disso, nesta fase, avançamos na conversão dos valores RGB para várias escalas de cinza, incluindo Média, Ponderada, Luminosidade, Dessaturação, bem como na Decomposição de Cores (máximo e mínimo). Este avanço amplia nossa capacidade de manipulação de imagens, permitindo uma gama mais ampla de efeitos e estilos visuais.