

```

%pip install opencv-python
%pip install matplotlib
import cv2
import numpy as np
import matplotlib.pyplot as plt

import cv2
import numpy as np
import matplotlib.pyplot as plt

# Função para carregar uma lista de imagens coloridas
def load_color_images(image_paths):
    images = [cv2.imread(path, cv2.IMREAD_COLOR) for path in
image_paths]
    return images

# Função para calcular a média de várias imagens coloridas
def average_color_images(images):
    # Converter a lista de imagens para um array numpy e calcular a
média por canal (RGB)
    average_image = np.mean(images, axis=0).astype(np.uint8) #
Corrigido para np.uint8
    return average_image

# Lista de caminhos das 5 imagens ruidosas coloridas
image_paths = [
    'passaros_R1.png',
    'passaros_R2.png',
    'passaros_R3.png',
    'passaros_R4.png',
    'passaros_R5.png'
]

# Carregar as imagens coloridas
images = load_color_images(image_paths)

# Calcular a imagem média (redução de ruído)
average_image = average_color_images(images)

# Converter a imagem de BGR (padrão OpenCV) para RGB (padrão
matplotlib)
average_image_rgb = cv2.cvtColor(average_image, cv2.COLOR_BGR2RGB) #
Corrigido para cv2.COLOR_BGR2RGB

# Exibir a imagem resultante
plt.figure(figsize=(6,6))
plt.title("Imagem Média Colorida (Redução de Ruído)")
plt.imshow(average_image_rgb)
plt.axis('off') # Ocultar os eixos
plt.show()

```

```
# Salvar a imagem resultante  
cv2.imwrite('average_color_image_result.jpg', average_image)
```

Imagem Média Colorida (Redução de Ruído)



True

```

%pip install opencv-python
%pip install matplotlib
import cv2
import numpy as np
import matplotlib.pyplot as plt

import cv2
import numpy as np
import matplotlib.pyplot as plt

# Função para carregar em escala de cinza ou coloridas
def load_images(image_path1, image_path2, color=False):
    if color:
        image1 = cv2.imread(image_path1, cv2.IMREAD_COLOR)
        image2 = cv2.imread(image_path2, cv2.IMREAD_COLOR)
    else:
        image1 = cv2.imread(image_path1, cv2.IMREAD_GRAYSCALE)
        image2 = cv2.imread(image_path2, cv2.IMREAD_GRAYSCALE)

    return image1, image2

# Função para subtrair as imagens
def subtract_images(image1, image2):
    # A função cv2.subtract lida com valores negativos adequadamente
    result = cv2.subtract(image1, image2)
    return result

# Caminhos das duas imagens
image_path1 = 'passaros.png'
image_path2 = 'Cinza.png'

# Carregar as imagens (em colorido ou escala de cinza)
# Use color=True se quiser subtrair imagens coloridas, senão deixa
# como False
image1, image2 = load_images(image_path1, image_path2, color=True)

# Verificar as dimensões das imagens
print(f"Dimensões da imagem 1: {image1.shape}")
print(f"Dimensões da imagem 2: {image2.shape}")

# Redimensionar a segunda imagem para que tenha as mesmas dimensões
# que a primeira, se necessário
if image1.shape != image2.shape:
    image2 = cv2.resize(image2, (image1.shape[1], image1.shape[0]))

# Subtrair as imagens
result_image = subtract_images(image1, image2)

# Exibir a imagem resultante
plt.figure(figsize=(6, 6))

```

```
plt.title("Resultado da Subtração de Imagens")
plt.imshow(cv2.cvtColor(result_image, cv2.COLOR_BGR2RGB))
plt.axis('off') # Ocultar os eixos
plt.show()

# Salvar a imagem resultante
cv2.imwrite('result_image.jpg', result_image)

# Exibir as imagens original e subtraída
plt.figure(figsize=(10, 6))

plt.subplot(1, 3, 1)
plt.title('Imagem 1')
plt.imshow(cv2.cvtColor(image1, cv2.COLOR_BGR2RGB))
plt.axis('off')

plt.subplot(1, 3, 2)
plt.title('Imagem 2')
plt.imshow(cv2.cvtColor(image2, cv2.COLOR_BGR2RGB))
plt.axis('off')

plt.subplot(1, 3, 3)
plt.title('Imagem Subtraída')
plt.imshow(cv2.cvtColor(result_image, cv2.COLOR_BGR2RGB)) # Corrigido
para result_image
plt.axis('off')

plt.tight_layout()
plt.show()

# Salvar a imagem resultante
cv2.imwrite('subtracted_image.jpg', result_image) # Corrigido para
result_image

Dimensões da imagem 1: (626, 455, 3)
Dimensões da imagem 2: (1024, 1024, 3)
```

## Resultado da Subtração de Imagens

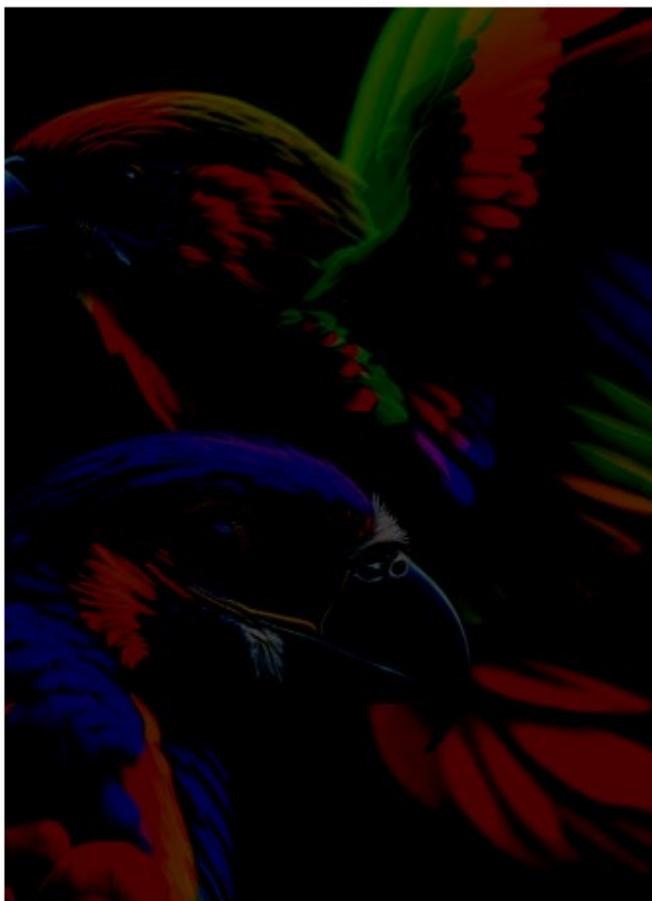


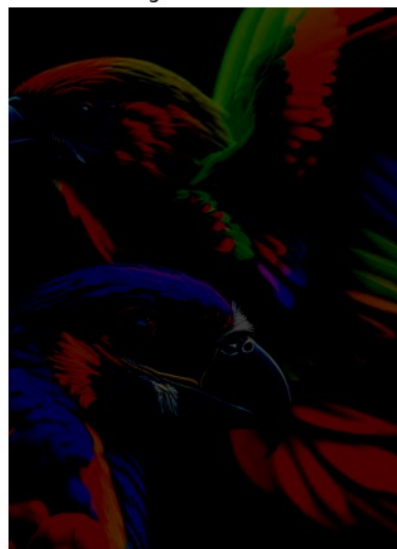
Imagem 1



Imagem 2



Imagem Subtraída



True

1) A média das imagens combina os pixels, fazendo com que o ruído se cancele e a imagem fique mais limpa. 2) O código não vai funcionar, porque as imagens precisam ter o mesmo tamanho. Você precisaria redimensionar a imagem menor. 3) A imagem resultante ficará toda preta (pixels com valor zero), e isso não causa problemas. 4) O OpenCV transforma esses valores negativos em zero, então a área correspondente ficará preta na imagem.