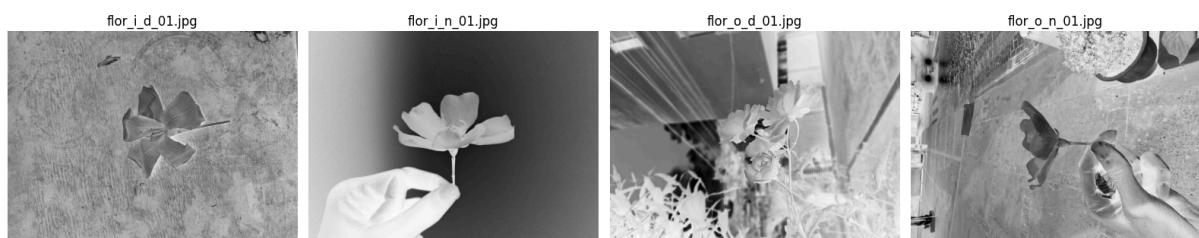
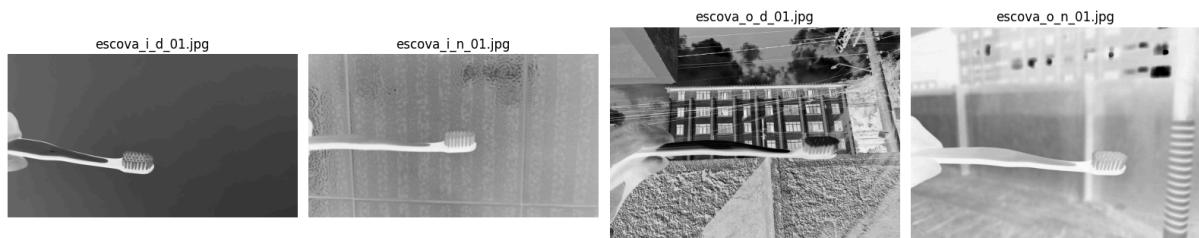
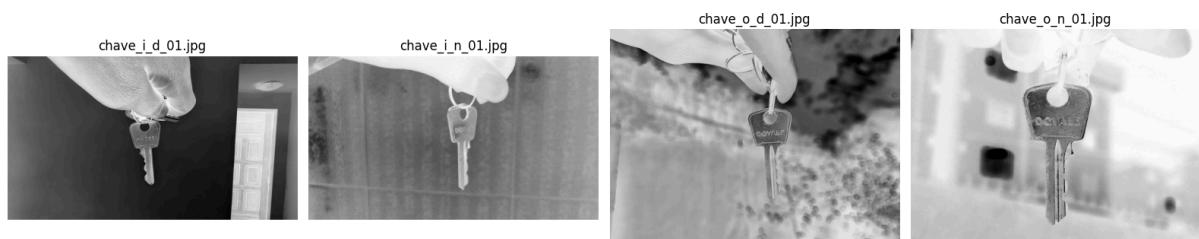
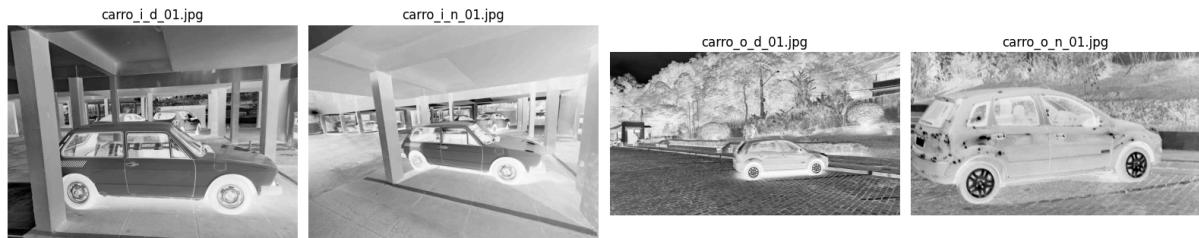


Atividade 4

Transformação 1 - Inversão de Cor



Carro

As imagens feitas durante o dia apresentam uma distribuição de iluminação mais ampla, com texturas bem definidas no ambiente, o que pode dificultar a etapa de segmentação para isolar o objeto de interesse. A imagem `carro_o_n_01`, por sua vez, sofre com reflexos intensos devido à iluminação artificial direta, resultando em saturação de pixels que podem ser interpretados erroneamente como bordas ou características inexistentes do veículo. Já a imagem `carro_i_n_01`, apesar da baixa luminosidade geral, a iluminação concentrada define contornos claros e reflexos que, embora possam ser ruídos, ajudam a delinear a curvatura do carro. O fundo de baixa complexidade simplifica consideravelmente a separação entre o objeto e o cenário, tornando-a uma candidata ótima para algoritmos de detecção de forma.

Chave

O conjunto de imagens da chave demonstra uma grande variabilidade em termos de contraste e nitidez. A imagem `chave_i_n_01` resultou em ganho de detalhes e bordas, que facilita a execução de algoritmo de extração de características em comparação à sua versão original. As imagens diurnas (`chave_i_d_01` e `chave_o_d_01`) oferecem excelentes condições de iluminação, com alto contraste e texturas bem definidas, ideais para a análise dos dentes da chave. No entanto, a imagem `chave_o_n_01` se destaca por uma combinação particular de fatores. A iluminação artificial noturna, focada no objeto, gera bordas de alto contraste contra um fundo com efeito que desfoca. Isso facilita a segmentação e permite que algoritmos de detecção de bordas operem com alta precisão no contorno serrilhado, uma característica fundamental para a identificação do objeto.

Escova

Para a classe "escova", o principal desafio é a relação entre o objeto e o fundo. Nas capturas externas (`escova_o_d_01` e `escova_o_n_01`), o fundo é complexo e texturizado, exigindo algoritmos de segmentação mais sofisticados para isolar a escova. A imagem `escova_i_n_01`, capturada em ambiente interno noturno, sofre de ruído de alta frequência e baixo contraste, fatores que são diminuídos na sua versão invertida. Porém, a imagem `escova_i_d_01` oferece condições quase ideais para o processamento. O fundo homogêneo e de alta iluminação cria um contraste acentuado com o objeto. A simplicidade do pré-processamento permite que a análise se concentre diretamente em características mais complexas do objeto, como a densidade e a orientação das cerdas.

Flor

O conjunto de imagens da "flor" apresenta variações significativas de cor, forma e iluminação. A imagem `flor_o_d_01` exibe múltiplas instâncias do objeto com oclusão parcial por folhas, um cenário complexo para um algoritmo de reconhecimento. A `flor_o_n_01` possui uma iluminação colorida não uniforme, com uma luz verde afetando a área inferior. A `flor_i_d_01` apresenta o objeto de forma plana e bem iluminada, ideal para análise de textura e cor das pétalas, mas com pouca informação sobre sua forma tridimensional, alta dificuldade de separação do fundo na sua versão invertida. A imagem `flor_i_n_01`, por outro lado, representa um caso particular e valioso. Devido à subexposição severa, a maior parte da informação de cor e textura é suprimida. O resultado é uma imagem de altíssimo contraste focada quase que exclusivamente na silhueta do objeto. Esta simplificação a torna extremamente adequada para algoritmos de análise de contorno.

Transformação 2 - Equalização



Carro

A imagem `carro_i_n_01` resulta na perda de detalhes em áreas de sombra. Ao aplicar a equalização, o objetivo é redistribuir essa intensidade para aumentar o contraste global. Na imagem `carro_i_n_01`, o efeito é notável: o algoritmo expande as cores, iluminando drasticamente as áreas escuras. Isso distingue mais facilmente o carro do ambiente à sua volta. A carroceria do carro ganha um novo nível de contraste em suas curvas, tornando-a um bom exemplo para a aplicação da técnica. Nas outras imagens, o efeito é mais sutil, pois suas distribuições de cor já eram mais bem equilibradas.

Chave

No conjunto da "chave", a versão em escala de cinza da imagem `chave_i_n_01` é extremamente escura, com pouquíssima informação visual discernível. A equalização, neste caso, ilustra uma falha comum de técnicas globais: como a maior parte da imagem é composta pela textura dos azulejos do fundo, o algoritmo otimiza o contraste para essa região. O resultado é um realce excessivo do fundo, enquanto a chave, que ocupa uma área menor, não é beneficiada na mesma proporção, tornando a imagem final de baixa qualidade para a análise do objeto. Em contrapartida, a imagem `chave_o_d_01` responde muito bem à equalização. O processo consegue equilibrar as áreas de alta intensidade (o céu e o muro) com as de média intensidade (a chave), resultando em um ganho de contraste local no objeto. A textura do metal e os detalhes dos seus dentes tornam-se mais evidentes sem degradar o restante da cena.

Escova

A classe "escova" expõe as limitações da equalização de histograma em cenários específicos. Nas imagens em escala de cinza, as capturas internas são caracterizadas por um fundo de cor sólida e uniforme que ocupa a maior parte da área da imagem. A equalização, por ser um método global, baseia sua transformação na distribuição de intensidade de todos os pixels. Consequentemente, o algoritmo tenta "esticar" o pico estreito do histograma correspondente ao fundo, o que frequentemente resulta na amplificação de ruído e na introdução de artefatos visuais (como gradientes falsos) nessa área. O objeto de interesse (a escova), por representar uma porção minoritária dos pixels, tem sua aparência pouco melhorada ou até degradada. Portanto, para todas as imagens desta classe, a técnica se mostrou inadequada, principalmente nas internas, onde o fundo foi realçado de forma negativa.

Flor

Para a classe "flor", a equalização produziu resultados variados. Na imagem [flor_i_d_01](#), a versão em escala de cinza já apresenta um bom contraste inicial entre a flor e a superfície escura. A aplicação da equalização melhora ainda mais essa distinção. O algoritmo empurra os tons de cinza da flor para níveis mais altos de intensidade e os do fundo para níveis mais baixos, aumentando a separação entre os dois. Isso não apenas faz a flor se destacar com mais vivacidade, mas também realça a textura sutil da superfície em que ela repousa, um resultado positivo. Nas outras imagens, como as noturnas, a equalização tende a amplificar o ruído já presente devido à baixa luminosidade, resultando em um ganho de clareza limitado.

Imagens Utilizadas

Github

processamento-de-imagens/images at master · andreb308/processamento-de-imagens
Contribute to andreb308/processamento-de-imagens development by creating an account on GitHub.
<https://github.com/andreb308/processamento-de-imagens/tree/master/images>

andreb308/
processamento-de-...

Contributor 1 Issues 0 Stars 0 Forks 0

Google Drive

grayscale - Google Drive
https://drive.google.com/drive/u/0/folders/1mWVvEW_MXpuPE_8nWBIGyuhLAFzXMp_9

Código Utilizado

Inversão de Cor

```
import matplotlib.pyplot as plt
import numpy as np
import os
from skimage import io, color, util

# Salvando os nomes de cada arquivo de imagem em uma lista
image_files = [f for f in os.listdir('./images') if os.path.isfile(os.path.join('./images', f))]
```

```

fig, axes = plt.subplots(4, 4, figsize=(16, 16))

for idx, filename in enumerate(image_files):
    filepath = os.path.join('./images', filename)
    image = io.imread(filepath)
    grayscale_img = color.rgb2gray(image)
    inverted_img = util.invert(grayscale_img)

    # plotando a imagem
    ax = axes[idx // 4, idx % 4]
    ax.imshow(inverted_img, cmap='gray')
    ax.set_title(filename)
    ax.axis('off')

plt.tight_layout()
plt.show()

```

Equalização

```

import matplotlib.pyplot as plt
import numpy as np
import os
from skimage import io, color, util
from skimage.exposure import equalize_hist

# Salvando os nomes de cada arquivo de imagem em uma lista
image_files = [f for f in os.listdir('./images') if os.path.isfile(os.path.join('./images', f))]
fig, axes = plt.subplots(4, 4, figsize=(16, 16))

for idx, filename in enumerate(image_files):
    filepath = os.path.join('./images', filename)
    image = io.imread(filepath)
    grayscale_img = color.rgb2gray(image)
    equalized_img = equalize_hist(grayscale_img)

    # plotando a imagem
    ax = axes[idx // 4, idx % 4]
    ax.imshow(grayscale_img, cmap='gray')
    ax.set_title(filename)
    ax.axis('off')

plt.tight_layout()
plt.show()

```