**Sistema de Processamento e Análise de Imagens**

**Instalação:**

pip install streamlit opencv-python pillow numpy matplotlib scikit-image scikit-learn reportlab

**Execução:**

streamlit run sistema\_processamento\_imagens.py

**Descrição Geral**

O **Sistema de Processamento e Análise de Imagens** é uma aplicação interativa desenvolvida com **Streamlit** e bibliotecas de visão computacional como **OpenCV** e **scikit-image**.  
Ele permite **carregar, processar, analisar e gerar relatórios** de imagens com métricas quantitativas de qualidade.

**Funcionalidades Principais**

**1. Upload e Normalização**

* Permite importar imagens **PNG ou JPEG** com até **10MB**.
* As imagens são automaticamente **redimensionadas para 512x512 pixels**.
* O sistema mantém tanto a imagem original quanto a processada em memória.

**2. Pré-Processamento**

* Filtros disponíveis:
  + **Gaussiano** – suaviza ruídos com controle de desvio padrão (*sigma*).
  + **Mediana** – remove ruídos preservando bordas.
* Parâmetros ajustáveis:
  + Raio do kernel (ímpar)
  + Sigma (0.5 a 2.0)

**3. Realce de Nitidez**

Aplica métodos para melhorar a definição da imagem:

* **Laplaciano** – realce via gradiente de segunda ordem.
* **Sobel** – realce baseado em gradiente direcional com limiar ajustável.
* **Alta Frequência** – enfatiza detalhes finos por subtração de componentes suaves.

Parâmetros:

* Peso (0.1–3.0)
* Limiar (para Sobel)
* Intensidade (para alta frequência)

**4. Realce de Contraste**

Dois modos de aprimoramento:

* **CLAHE (Local)** – equalização adaptativa com controle de *clip limit* e tamanho de bloco.
* **Equalização Global** – redistribui os níveis de intensidade para todo o histograma.

Parâmetros:

* *Clip Limit* (2.0–3.0)
* Tamanho do bloco (4, 8 ou 16)

**5. Cálculo de Métricas**

Após o processamento, o sistema calcula métricas quantitativas:

* **PSNR (Peak Signal-to-Noise Ratio)** – mede relação sinal/ruído (ideal ≥ 30 dB)
* **SSIM (Structural Similarity Index)** – avalia similaridade estrutural (ideal ≥ 0.85)
* **LC (Local Contrast)** – mede contraste local
* **Edge Sharpness** – quantifica a nitidez das bordas

Essas métricas são exibidas em formato de **cards** e **gráficos comparativos**.

**6. Análise Visual**

* Exibe **comparações lado a lado** entre imagem original e processada.
* Mostra:
  + Mapa de diferença entre as imagens.
  + Bordas detectadas (Canny).
  + Histogramas RGB comparativos.

**7. Relatórios e Exportação**

* Gera **relatório PDF automático** com:
  + Imagens original e processada.
  + Métricas calculadas.
  + Histórico de operações realizadas.
* Permite **baixar a imagem processada em formato PNG**.

**8. Histórico e Controle**

* Cada ação é registrada com **data e hora** no histórico.
* Funções de **limpar histórico** e **resetar o sistema**.
* Controle de usuário (Operador / Administrador).

**9. Interface**

* Desenvolvida com **Streamlit**, possui:
  + Abas para upload, processamento, análise, métricas e relatório.
  + Layout responsivo e estilização customizada em CSS.
  + Feedback visual (mensagens, balões e indicadores).

**10. Relatório Técnico**

O relatório PDF inclui:

* Informações do usuário e data.
* Imagens originais e processadas.
* Métricas quantitativas (PSNR, SSIM, LC, Edge Sharpness).
* Histórico das operações realizadas.
* Formatação automática no padrão **A4**, gerada com **ReportLab**.

**Tecnologias Utilizadas**

* **Python 3.x**
* **Streamlit** – interface interativa
* **OpenCV** – processamento de imagens
* **scikit-image** – filtros e métricas
* **Matplotlib** – visualizações e histogramas
* **ReportLab** – geração de relatórios PDF
* **Pillow (PIL)** – manipulação de imagens

**Estrutura de Abas**

1. **📤 Upload** – Importar imagem
2. **🔧 Processamento** – Aplicar filtros e realces
3. **📊 Análise** – Comparação e histogramas
4. **📈 Métricas** – Avaliação quantitativa
5. **📄 Relatório** – Exportar resultados e histórico