

# PDI Interativo: Uma Ferramenta para Estudo e Experimentação com Abordagem Espacial e Frequencial

João Lucas Mayrinck<sup>1</sup>

<sup>1</sup>Instituto de Ciências Exatas e Tecnológicas – Universidade Federal de Viçosa – Campus Rio Paranaíba

Caixa Postal 22 – 38.810-000 – Rio Paranaíba – MG – Brasil

joao.d@ufv.br

**Abstract.** This article presents an interactive system for digital image processing (DIP), focusing on the application and analysis of spatial and frequency domain filters. The system, developed in Python (Streamlit) with a complementary web interface in HTML/JavaScript, aims to serve as an educational and research tool. It allows image manipulation using low-pass filters and high-pass filters, as well as frequency domain operations via the Fourier Transform. The system also includes features for adding and removing various types of noise, segmentation, and morphological operations. The results demonstrate the tool's effectiveness in visualizing filter effects and understanding DIP concepts, although it presents performance limitations with high-resolution images and lacks automatic quantitative metrics. The work concludes by highlighting the relevance of the tool for teaching and research in DIP, suggesting future improvements such as the inclusion of quality metrics, support for multiple formats, and integration with machine learning.

**Keywords:** Digital Image Processing, Spatial Filters, Frequency Domain, Educational Tool, Morphological Operations, Image Segmentation.

**Resumo.** Este artigo apresenta um sistema interativo para processamento digital de imagens (PDI), focado na aplicação e análise de filtros espaciais e freqüenciais. O sistema, desenvolvido em Python (Streamlit) e com uma interface web complementar em HTML/JavaScript, visa ser uma ferramenta didática e de pesquisa. Ele permite a manipulação de imagens através de filtros passa-baixa e passa-alta, além de operações no domínio da frequência via Transformada de Fourier. O sistema também inclui funcionalidades para adição e remoção de diversos tipos de ruído, segmentação e operações morfológicas. Os resultados demonstram a eficácia da ferramenta na visualização dos efeitos dos filtros e na compreensão dos conceitos de PDI, embora apresente limitações de desempenho para imagens de alta resolução e ausência de métricas quantitativas automáticas. O trabalho conclui sobre a relevância da ferramenta para o ensino e pesquisa em PDI, sugerindo futuras melhorias como a inclusão de métricas de qualidade, suporte a múltiplos formatos e integração com aprendizado de máquina.

**Palavras-Chave:** Processamento Digital de Imagens, Filtros Espaciais, Domínio da Frequência, Ferramenta Educacional, Operações Morfológicas, Segmentação de Imagens.

## **1. Introdução**

O processamento digital de imagens (PDI) é uma área fundamental da ciência da computação e da engenharia, com aplicações que abrangem desde a medicina e segurança até o entretenimento e o sensoriamento remoto [Gonzalez & Woods, 2018]. A capacidade de manipular e analisar imagens digitalmente permite a extração de informações valiosas, a melhoria da qualidade visual e a automação de tarefas complexas. A relevância do PDI está em sua ampla aplicabilidade e no impacto transformador que exerce em diversas disciplinas, tornando-se uma ferramenta indispensável para a compreensão e interação com o mundo visual [Marques, 2014].

O PDI possui aplicações diversas que vão do ensino à pesquisa aplicada e a setores profissionais. No âmbito educacional, permite que estudantes experimentem conceitos fundamentais de forma prática e visual. Na pesquisa, favorece a prototipagem rápida de algoritmos e a análise exploratória de conjuntos de imagens. Já em áreas técnicas e biomédicas, é essencial em tarefas como remoção de ruído, realce de características e segmentação de regiões de interesse [Pedrini & Schwartz, 2008].

Diante desse cenário, este trabalho apresenta o desenvolvimento de um sistema interativo para processamento digital de imagens. A ferramenta foi projetada para aplicar filtros espaciais e freqüenciais em imagens digitais, possibilitando a análise visual dos efeitos resultantes dessas operações. Seu objetivo principal é oferecer uma plataforma intuitiva e didática, que permita a usuários de diferentes níveis de conhecimento explorar fundamentos do PDI, compreender a interação entre os domínios espacial e da freqüência e visualizar, de forma prática, o impacto de diferentes técnicas de filtragem.

## **2. Referencial Teórico**

O processamento digital de imagens baseia-se em uma série de conceitos matemáticos e algorítmicos que permitem a manipulação de dados visuais. Dois domínios principais são explorados para essa finalidade: o domínio espacial e o domínio da freqüência. Cada um oferece abordagens distintas para o tratamento de imagens, com vantagens específicas dependendo da aplicação desejada [Forsyth & Ponce, 2011].

### **2.1. Filtros Espaciais**

Filtros espaciais operam diretamente nos *pixels* da imagem, modificando o valor de cada pixel com base nos valores de seus vizinhos. Essa operação é tipicamente realizada através de uma convolução, onde uma pequena matriz, conhecida como *kernel* ou máscara, é aplicada sobre a imagem. Os filtros espaciais podem ser classificados em passa-baixa e passa-alta, dependendo do efeito desejado [Lima et al., 2015].

### **2.2. Transformada de Fourier e Filtros no Domínio da Frequênci**a

A Transformada de Fourier (TF) é uma ferramenta matemática que decompõe uma imagem do domínio espacial para o domínio da freqüência. No domínio da freqüência, a imagem é representada por suas componentes de freqüência, onde as baixas freqüências

correspondem a variações lentas de intensidade e as altas frequências correspondem a variações rápidas [Gonçalves, 2004].

A filtragem no domínio da frequência envolve a modificação do espectro de Fourier da imagem e, em seguida, a aplicação da Transformada Inversa de Fourier (TIF) para retornar a imagem ao domínio espacial. Isso permite um controle preciso sobre quais componentes de frequência são realçadas ou atenuadas [Benício, 2022].

### 3. Trabalhos Relacionados

Diversas ferramentas educacionais e plataformas interativas têm sido desenvolvidas para facilitar o aprendizado e a experimentação em Processamento Digital de Imagens. Essas ferramentas variam em escopo, funcionalidades e tecnologias empregadas, cada uma com suas vantagens e desvantagens.

Por exemplo, o ImageJ [Schneider, Rasband & Eliceiri, 2012], amplamente usado na análise científica, destaca-se pela extensibilidade e comunidade ativa, mas exige curva de aprendizado elevada. Ferramentas *web*, como o OpenCV.js [Bradski, 2000], permitem a execução de algoritmos de PDI diretamente no navegador, eliminando a necessidade de instalações complexas. A vantagem é a acessibilidade e a portabilidade, mas podem ter limitações de desempenho para operações mais intensivas ou com imagens de alta resolução. Plataformas como o MATLAB Image Processing Toolbox [Eddins, 2004] oferecem um ambiente robusto para pesquisa e desenvolvimento em PDI, com funções otimizadas e vasta documentação. Contudo, o custo da licença e a necessidade de conhecimento em programação MATLAB podem ser barreiras para estudantes e entusiastas.

Nesse cenário, o presente trabalho busca preencher uma lacuna ao oferecer interface intuitiva e didática, unindo bibliotecas Python otimizadas (via Streamlit) e acessibilidade *web* (HTML/JavaScript), com foco em visualização em tempo real e aprendizado prático de conceitos fundamentais, mas sem a complexidade das ferramentas tradicionais.

### 4. Metodologia

O desenvolvimento do sistema interativo para processamento digital de imagens seguiu uma abordagem modular, visando facilitar a implementação, manutenção e escalabilidade das funcionalidades. A arquitetura do projeto foi concebida para integrar diferentes componentes, desde a interface do usuário até os algoritmos de processamento de imagem, utilizando tecnologias amplamente reconhecidas na área.

#### 4.1. Linguagens e Ferramentas Utilizadas

O projeto foi implementado utilizando Python como linguagem de programação principal, aproveitando sua vasta gama de bibliotecas para computação científica e processamento de imagens. As principais bibliotecas empregadas incluem:

- **OpenCV:** Essencial para operações de leitura, escrita e manipulação de imagens, bem como para funções de processamento de baixo nível;

- **NumPy:** Fundamental para operações numéricas eficientes, especialmente para manipulação de vetores multidimensionais que representam as imagens;
- **Matplotlib:** Utilizada para a geração de gráficos e visualizações, como histogramas;
- **Streamlit:** Uma estrutura de código aberto que permitiu a criação de uma interface de usuário interativa e amigável para a aplicação Python, facilitando a experimentação com os filtros e operações.

Além da aplicação Streamlit, uma interface *web* complementar foi desenvolvida utilizando HTML, CSS e JavaScript. Esta interface demonstra a portabilidade e a capacidade de execução das operações de processamento diretamente no navegador, utilizando a API Canvas para manipulação de *pixels* e implementações em JavaScript de algoritmos.

## 4.2. Implementação das Operações

### 4.2.1. Filtros Espaciais

Os filtros espaciais operam diretamente nos pixels da imagem por meio de convolução com um kernel. Para filtros não lineares como a mediana, a operação envolve a ordenação dos valores da vizinhança e a seleção do valor central. Os principais exemplos incluem filtros passa-baixa (suavização) e passa-alta (realce de bordas) [Gonzalez & Woods, 2018]. A Figura 1 resume alguns kernels clássicos amplamente utilizados no ensino de PDI e seus efeitos nas imagens.

Filtro	Kernel (3x3)	Efeito principal
Média	$\begin{bmatrix} 1/9, 1/9, 1/9 \\ 1/9, 1/9, 1/9 \\ 1/9, 1/9, 1/9 \end{bmatrix}$	Suavização uniforme
Gaussiano	Aproximação discreta	Suavização com preservação parcial de bordas
Sobel X	$\begin{bmatrix} -1, 0, 1 \\ -2, 0, 2 \\ -1, 0, 1 \end{bmatrix}$	Detecção de bordas verticais
Sobel Y	$\begin{bmatrix} -1, -2, -1 \\ 0, 0, 0 \\ 1, 2, 1 \end{bmatrix}$	Detecção de bordas horizontais
Laplaciano	$\begin{bmatrix} 0, -1, 0 \\ -1, 4, -1 \\ 0, -1, 0 \end{bmatrix}$	Realce de bordas em todas direções

**Figura 1: Exemplos de kernels clássicos em PDI**

A implementação dos filtros espaciais baseou-se na operação de convolução 2D. Para cada pixel da imagem de entrada, um *kernel* (máscara) é aplicado sobre sua vizinhança. O novo valor do *pixel* de saída é calculado como a soma ponderada dos *pixels* da vizinhança, onde os pesos são definidos pelo *kernel*.

#### 4.2.2. Filtros Frequenciais

As operações no domínio da frequência seguem os seguintes passos:

- **Transformada de Fourier:** A imagem de entrada (em tons de cinza) é transformada para o domínio da frequência utilizando a Transformada Rápida de Fourier. Isso resulta em um espectro complexo que contém informações de magnitude e fase;
- **Shift do Espectro:** O espectro de Fourier é rearranjado (*shifted*) para que a componente de frequência zero (DC) seja centralizada. Isso facilita a visualização e a aplicação de máscaras de filtro;
- **Aplicação de Máscaras:** São criadas máscaras no domínio da frequência baseadas nos *kernels* tradicionais utilizados no domínio espacial. Essas máscaras são multiplicadas ponto a ponto com o espectro de Fourier da imagem;
- **Transformada Inversa de Fourier:** O espectro filtrado é transformado de volta para o domínio espacial utilizando a Transformada Inversa de Fourier, resultando na imagem processada.

#### 4.2.3. Interface Interativa via HTML e Streamlit

A interatividade é um pilar central do sistema. A aplicação criada utilizando Streamlit oferece uma interface de usuário intuitiva com:

- **Upload de Imagens:** Permite que o usuário carregue suas próprias imagens para processamento;
- **Visualização em Tempo Real:** Exibe a imagem original, em tons de cinza, seus histogramas e os resultados dos filtros aplicados em tempo real;
- **Seleção de Filtros e Parâmetros:** *Menus* suspensos e *sliders* permitem que o usuário escolha o tipo de filtro e ajuste seus parâmetros;
- **Pipeline de Operações:** Uma funcionalidade de *pipeline* permite que os usuários encadeiem múltiplas operações sequencialmente, observando o efeito cumulativo de cada etapa. Isso é particularmente útil para fluxos de trabalho complexos de PDI;
- **Download de Resultados:** Possibilita o *download* das imagens processadas e dos histogramas.

A interface *web* — criada com HTML, CSS e JavaScript — complementa a aplicação Streamlit, oferecendo uma demonstração leve e acessível das funcionalidades de filtragem e análise espectral diretamente no navegador, sem a necessidade de um ambiente Python configurado, replicando a lógica de convolução e TF, proporcionando uma experiência interativa similar.

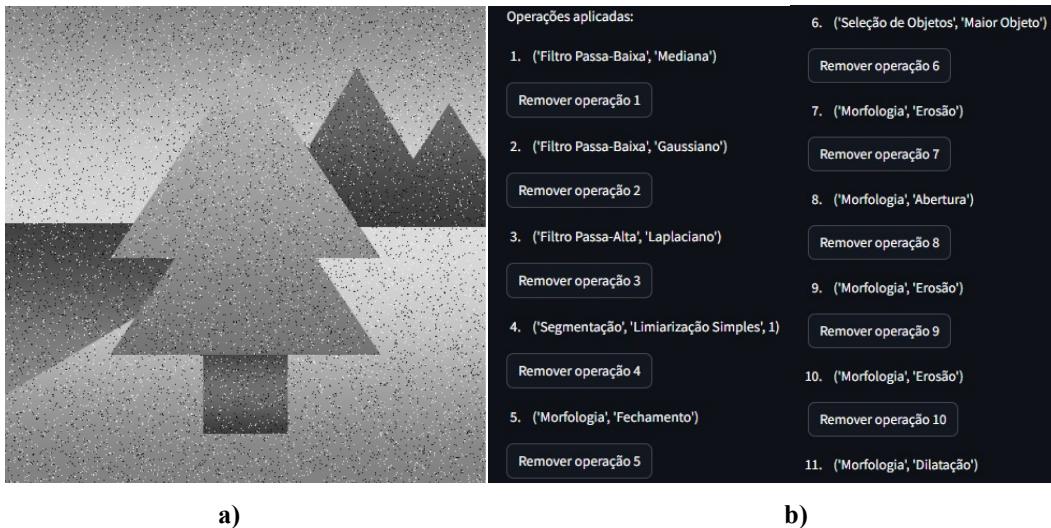
### 5. Resultados

O sistema interativo desenvolvido demonstrou ser uma ferramenta eficaz para a visualização e experimentação de diversas técnicas de processamento digital de imagens. Através de sua interface intuitiva, foi possível aplicar filtros espaciais e freqüenciais,

adicionar diferentes tipos de ruído e observar o impacto dessas operações na qualidade e nas características visuais das imagens.

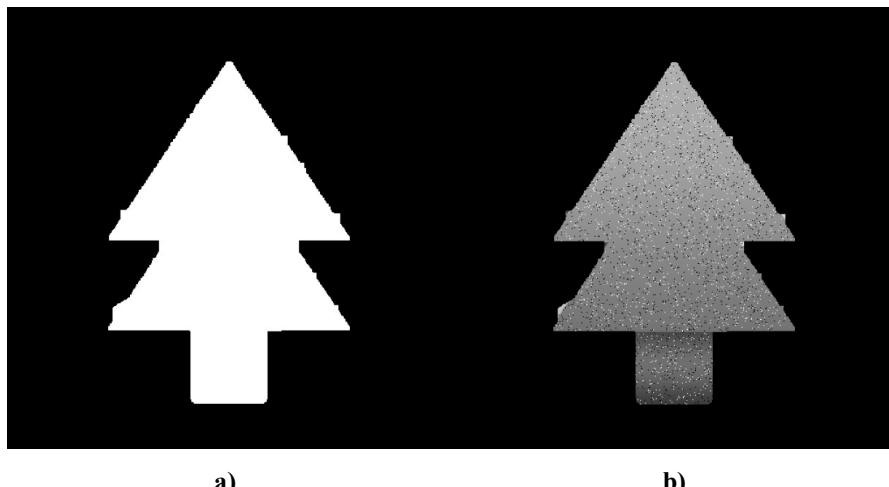
### 5.1. Exemplos de Aplicação

Para ilustrar a funcionalidade do sistema, utilizamos a Figura 1.a como imagem ruidosa de exemplo e aplicamos uma série de operações conforme ilustrado pela Figura 1.b.



**Figura 1: a) Imagem de exemplo e b) Pipeline de operações**

Após a aplicação das operações descritas no pipeline presente na Figura 1.b sobre a imagem de exemplo ilustrada na Figura 1.a, a imagem resultante é numa máscara binária representada na Figura 2.a e ao aplicá-la sobre a imagem original, obtemos o resultado ilustrado pela Figura 2.b.

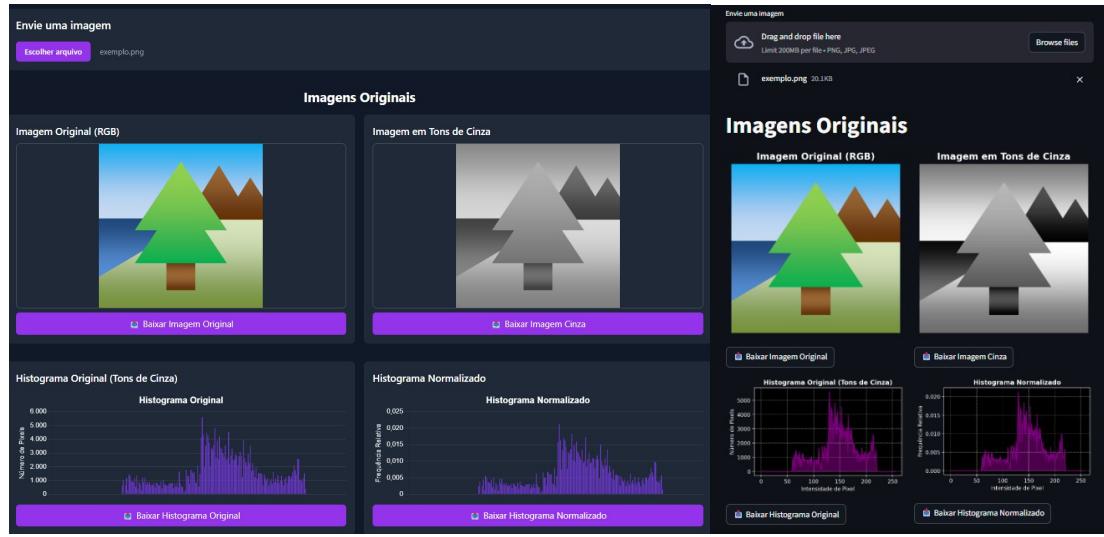


**Figura 2: a) Imagem de exemplo e b) Pipeline de operações**

### 5.2. Telas da Interface

A interface do usuário, desenvolvida em Streamlit e HTML/JavaScript, oferece uma experiência interativa e visualmente rica, sendo composta das seguintes telas:

- **Tela Principal:** Com as seções de *upload* de imagem, exibição da imagem original (RGB e tons de cinza) e seu histograma original e normalizado, conforme ilustrado na Figura 3;



a)

b)

**Figura 3: Tela Principal a) Interface HTML e b) Interface Streamlit**

- **Seção de Descritores:** Exibe os resultados da análise de descritores de cor, forma e textura, incluindo o histograma LBP para textura, conforme demonstrado na Figura 4;



a)

b)

**Figura 4: Seção de Descritores a) Interface HTML e b) Interface Streamlit**

- **Controles de Operações e *Preview*:** *Menus suspensos* para seleção de operações e filtros (clássicos e personalizados) e os *sliders* para ajuste de parâmetro e exibindo o resultado da operação selecionada antes de ser adicionada ao pipeline, conforme exibido na Figura 5;



a)

b)

**Figura 5: Operações e *Preview* a) Interface HTML e b) Interface Streamlit**

- **Pipeline de Operações:** Uma lista das operações adicionadas, com a opção de remover cada uma, e a exibição da imagem resultante do *pipeline*, conforme observado na Figura 6;

**Pipeline de Operações**

lowpass (gaussian)

highpass (laplacian)

segmentation (simple) [limiar: 1]

**Limpar Pipeline**

Operações aplicadas:

- ('Filtro Passa-Baixa', 'Gaussiano')
- ('Filtro Passa-Alta', 'Laplaciano')
- ('Segmentação', 'Limiarização Simples', 1)

Remover operação 1

Remover operação 2

Remover operação 3

Limpar Pipeline

a)

b)

Figura 6: Pipeline de Operações a) Interface HTML e b) Interface Streamlit

- **Aplicação de Máscara:** A funcionalidade de aplicar o resultado do *pipeline* como uma máscara sobre a imagem original, útil para segmentação e isolamento de regiões de interesse, conforme ilustrado pela Figura 7.

**Aplicar Máscara na Imagem Original**

Aplicar Máscara na Imagem Original

**Imagen Original com Máscara Aplicada**

a)

b)

Figura 7: Aplicação de Máscara a) Interface HTML e b) Interface Streamlit

### 5.3. Implantação

Visando democratizar o acesso à ferramenta, o sistema — juntamente com seu código fonte — está disponível no GitHub, por onde é possível contribuir com o projeto, e sua interface *web* pode ser acessada pela página [jluckmay.github.io/Processamento-Digital-de-Imagens](https://jluckmay.github.io/Processamento-Digital-de-Imagens).

## 6. Discussão

O sistema interativo para processamento digital de imagens desenvolvido representa uma ferramenta robusta e didática, que oferece uma plataforma acessível para a exploração de conceitos fundamentais em PDI. A discussão a seguir aborda os pontos fortes e as limitações observadas durante o desenvolvimento e a experimentação com o sistema.

### 6.1. Pontos Fortes

Um dos principais pontos fortes do sistema é sua interface altamente interativa e intuitiva, tanto na versão Streamlit quanto na versão *web* em HTML/JavaScript. A capacidade de carregar imagens, selecionar filtros e ajustar parâmetros em tempo real, com *feedback* visual imediato, facilita enormemente a compreensão dos efeitos de cada operação. Isso o torna uma ferramenta ideal para fins educacionais, permitindo que estudantes e pesquisadores experimentem e visualizem conceitos complexos de PDI de forma prática.

O sistema abrange uma ampla gama de filtros espaciais e frequenciais, funcionalidades para adição e remoção de diversos tipos de ruído, segmentação e operações morfológicas. Essa abrangência permite explorar um vasto espectro de aplicações de PDI dentro de uma única plataforma. A funcionalidade de pipeline é um diferencial didático significativo. Ela permite que os usuários construam sequências complexas de operações, observando como cada etapa afeta o resultado final. Isso é crucial para entender a interdependência das técnicas de PDI e para desenvolver fluxos de trabalho otimizados para problemas específicos. A inclusão de diferentes tipos de ruído e a possibilidade de aplicar filtros para sua remoção também reforçam o aspecto experimental do sistema, permitindo a avaliação prática da eficácia de cada método.

Embora não seja o foco principal do sistema, a capacidade de calcular e exibir descritores de imagem adiciona uma camada de análise quantitativa e complementa a visualização, fornecendo informações numéricas sobre as características da imagem, o que é valioso para a análise aprofundada de imagens digitais. A implementação em HTML/JavaScript demonstra a viabilidade de executar operações de PDI diretamente no navegador, sem a necessidade de um servidor *backend* Python. Isso aumenta a acessibilidade do sistema, permitindo que seja utilizado em praticamente qualquer dispositivo com um navegador *web* moderno.

## 6.2. Limitações

Embora o sistema funcione bem para imagens de tamanho moderado, o desempenho pode ser um gargalo para imagens de alta resolução. As operações de convolução e Transformada de Fourier, especialmente as implementadas em JavaScript, podem consumir recursos computacionais significativos, levando a tempos de processamento mais longos. A versão Streamlit, que utiliza bibliotecas Python otimizadas (OpenCV, NumPy), tende a ser mais eficiente, mas ainda pode enfrentar desafios com imagens muito grandes. O sistema não incorpora métricas de avaliação de qualidade de imagem automatizadas, como PSNR (*Peak Signal-to-Noise Ratio*) ou SSIM (*Structural Similarity Index Measure*). A avaliação da eficácia dos filtros e da remoção de ruído é predominantemente qualitativa (visual). A inclusão dessas métricas forneceria uma base mais objetiva para a comparação e otimização de algoritmos. O sistema foca principalmente em imagens em tons de cinza para a maioria das operações de filtragem e Fourier. Embora suporte imagens RGB na entrada, a conversão para tons de cinza é uma etapa comum. A expansão para processamento direto em múltiplos canais para todas as operações poderia enriquecer a funcionalidade.

## 7. Conclusão e Trabalhos Futuros

O sistema interativo desenvolvido cumpriu o objetivo de oferecer uma plataforma acessível para experimentação em PDI, permitindo aplicar filtros espaciais e freqüenciais de forma prática e didática. A visualização em tempo real e o uso de

pipelines de operações mostraram-se eficazes para o ensino e compreensão de conceitos fundamentais.

Apesar de limitações em imagens de alta resolução e no processamento multicanal, soluções como paralelização em GPU, redução de dimensionalidade e a inclusão de métricas quantitativas (PSNR, SSIM) configuram caminhos promissores para ampliar a robustez do sistema. Assim, a ferramenta tem potencial para evoluir como recurso educacional e de pesquisa cada vez mais relevante.

## Referências

- Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing (4th ed.). Pearson.
- Marques, O. (2014). Processamento Digital de Imagens: Uma Abordagem Prática com MATLAB. Brasport.
- Pedrini, H., & Schwartz, W. R. (2008). Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações. Cengage Learning.
- Forsyth, D. A., & Ponce, J. (2011). Computer Vision: A Modern Approach (2nd ed.). Pearson.
- Lima, L. C. et al. (2015). Verificação de modelos aplicada aos filtros espaciais em processamento de imagens. ENCOSIS.
- Gonçalves, L. A. (2004). Um estudo sobre a Transformada Rápida de Fourier e seu uso em processamento de imagens. Universidade Federal do Rio Grande do Sul.
- Benício, B. G. S. (2022). Uma ferramenta didática para ensino de filtragem no domínio da frequência. Universidade Federal do Rio Grande do Norte.
- Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, 9(7), 671–675.
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Eddins, S. L. (2004). Digital Image Processing Using MATLAB. MathWorks & Gatesmark Publishing.