

UNIT

Universidade Tiradentes

Spectrum: Ferramenta de Análise e Correção de Imagens

Cauan Lemos Barbosa Paulo

Leandro Dias Silva

Paulo Souza Telles Filho

Erika Dias do Nascimento Marques

Thiago Vinny Horta França Menezes

Aracaju, novembro 2025

Cauan Lemos Barbosa Paulo, Leandro Dias Silva, Paulo Souza Telles Filho, Erika
Dias do Nascimento Marques, Thiago Vinny Horta França Menezes.

Spectrum: Ferramenta de Análise e Correção de Imagens

Aracaju, novembro 2025

Sumário

1. Introdução
2. Objetivos
3. Metodologia
 - 3.1 Estrutura do Sistema
 - 3.2 Funcionalidades
4. Fluxo de Execução
5. Resultados Esperados
6. Conclusão
7. Referências

1. Introdução

O projeto Spectrum é uma ferramenta interativa desenvolvida para análise, processamento e correção digital de imagens. Sua proposta consiste em reunir, em um único módulo, recursos de inspeção, transformação linear e não linear, equalização e visualização comparativa, fornecendo ao usuário controle total sobre o fluxo de edição e permitindo rastreabilidade completa das operações executadas. O sistema foi pensado e desenvolvido em Python, utilizando OpenCV, NumPy, Pillow e Matplotlib, integrados a um módulo central que gerencia todas as etapas do processamento.

Este documento apresenta os objetivos, a metodologia adotada, as imagens utilizadas nos testes e os resultados preliminares obtidos.

2. Objetivos do Projeto

O módulo principal do Spectrum tem como objetivo:

1. Integrar as funções de carregamento, processamento e análise de imagens, permitindo ao usuário realizar transformações com controle direto dos parâmetros.
2. Aplicar técnicas de processamento digital, lineares e não lineares, para corrigir brilho, contraste e distribuição tonal.
3. Oferecer mecanismos de avaliação, como histogramas comparativos e visualização lado a lado.
4. Registrar automaticamente todas as operações, garantindo reprodutibilidade e transparência no fluxo de edição.
5. Disponibilizar uma interface baseada em menu, que conduz o usuário por todas as etapas de forma simples e estruturada.

3. Metodologia

A metodologia adotada envolve um fluxo sequencial controlado, no qual cada etapa depende da conclusão anterior:

3.1 Importação da Imagem

O sistema lista automaticamente os arquivos disponíveis na pasta *img*, o usuário seleciona a imagem desejada. A imagem é convertida para o formato RGB \rightarrow BGR (padrão OpenCV), e por fim, a operação é registrada no histórico.

3.2 Parametrização

O usuário define os parâmetros que serão utilizados no processamento, como brilho, contraste, gamma, tipo de transformação não linear (logarítmica ou gama) e o método de equalização (global ou adaptativa). A parametrização garante flexibilidade e controle sobre o resultado final.

3.3 Processamento Linear

Aplicado conforme a função:

$$I'(x,y) = \alpha \cdot I(x,y) + \beta$$

Onde:

- $\alpha \rightarrow$ contraste
- $\beta \rightarrow$ brilho

É utilizado para ajustes globais de exposição.

3.4 Processamento Não Linear

Inclui: Correção gama: realce específico de regiões escuras ou claras, e função logarítmica: expansão de regiões de baixa intensidade. Ambas operam com normalização entre 0 e 1, garantindo precisão.

3.5 Equalização de Histograma

Dois métodos implementados: Global (cv2.equalizeHist) e adaptativa (CLAHE) para correções mais localizadas onde a equalização é aplicada no canal Y do espaço YCrCb para preservar coloração.

3.6 Análise de Histograma

Geração de histogramas RGB da imagem: Original e processada, permitindo avaliar a redistribuição tonal após as operações realizadas.

3.7 Visualização Comparativa

Gera imagem fusionada usando ponderação:

$$\textit{blended} = \alpha \cdot \textit{processed} + (1 - \alpha) \cdot \textit{original}$$

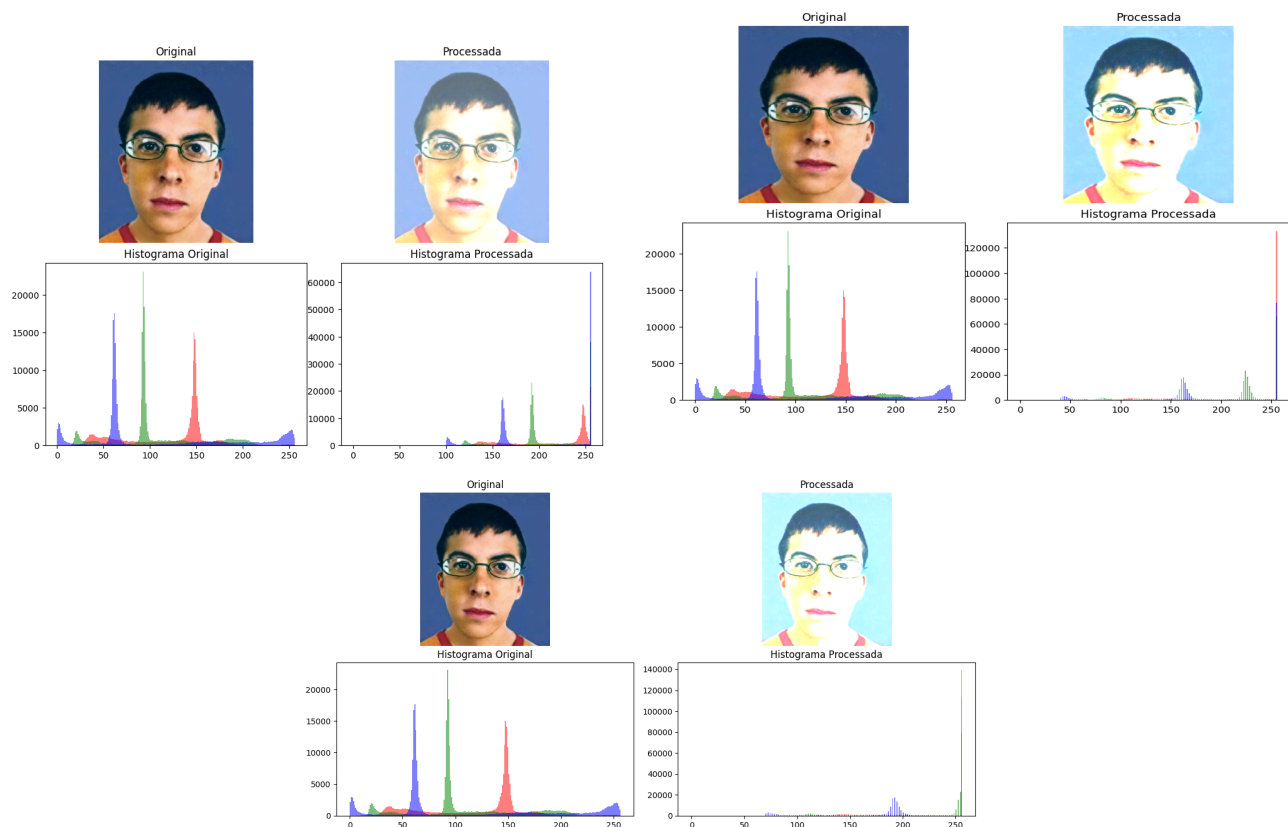
O usuário escolhe o nível de mistura.

3.8 Registro e Rastreamento

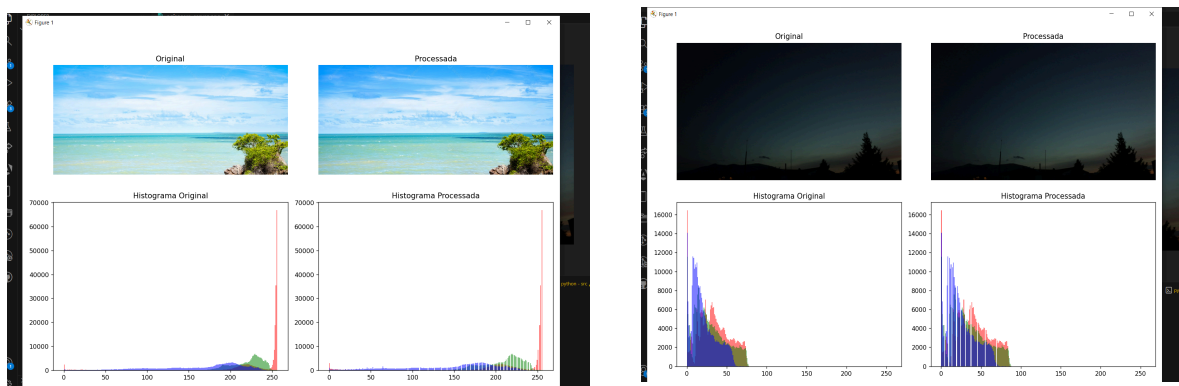
Cada operação é registrada com: o tipo de operação, parâmetros utilizados, data e hora e perfil de usuário, por fim, o relatório final é exportado em JSON.

4. Imagens Utilizadas nos Testes

Durante o desenvolvimento, foram utilizadas três imagens diferentes armazenadas na pasta img. Nas figuras abaixo, observamos exemplos de: histograma original e processado e imagem linear e não linear.



Figuras 1, 2 e 3: Provenientes dos testes do módulo Transformações Não-Lineares e Equalização de Histograma.



Figuras 4 e 5: Provenientes dos testes do módulo Transformações Não-Lineares e Linear.

5. Resultados Obtidos

5.1 Transformações Lineares

O ajuste de brilho e contraste produziu mudanças proporcionais nos níveis de intensidade, e a visualização dos histogramas revelou deslocamentos horizontais (brilho) e estiramento/compressão (contraste).

5.2 Transformações Não-Lineares

A correção gama destacou-se por realçar detalhes em regiões de sombra, a transformação logarítmica foi útil em imagens com pouca variação tonal. E em ambos os casos, os histogramas apresentaram redistribuição não uniforme, confirmando a operação.

5.3 Equalização de Histograma

O método global gerou forte expansão tonal, O método CLAHE produziu melhorias mais equilibradas, especialmente em regiões sensíveis como pele e sombras.

5.4 Comparação Visual

A comparação lado a lado permitiu observar: Recuperação de detalhes, melhora de contraste e também eventuais saturações quando parâmetros extremos foram usados.

5.5 Registro Automatizado

O relatório JSON demonstra: o número total de operações realizadas, os tipos mais frequentes e a sequência cronológica das modificações, evidenciando assim o caráter rastreável do sistema.

6. Conclusão Parcial

O módulo Spectrum apresenta um conjunto sólido de funcionalidades para análise e correção de imagens. As operações lineares e não lineares atuam de maneira complementar, permitindo ao usuário alcançar resultados precisos conforme os requisitos da imagem processada.

O sistema demonstra: Robustez na execução das transformações, clareza na interface por menu, transparência via registro de operações, eficiência na visualização e comparação dos resultados. Assim cumprindo com as funções pré-definidas e agindo de acordo com o esperado.

7. Referências Técnicas

- Documentação OpenCV.
- NumPy API Reference.