Universidade Católica Dom Bosco  
Engenharia de Computação

Luan Henrique Santos Miranda   
João Henrique Schweitzer Rezende

Filtros e Ruídos

Campo Grande – MS

2024

Luan Henrique Santos Miranda  
João Henrique Schweitzer Rezende

Filtros e Ruídos

Projeto desenvolvido para apresentação na disciplina Visão Computacional do curso de Engenharia da Computação, Universidade Católica Dom Bosco.  
  
Professor: Hemerson Pistori

Campo Grande – MS

2024

**Relatório de Desenvolvimento – Filtros e Ruidos**

**Introdução**

Este projeto tem como objetivo desenvolver um conjunto de filtros para processamento de imagens, permitindo a aplicação de diferentes tipos de ruído e suavização, além de técnicas de detecção de bordas. Através do uso de bibliotecas populares de processamento de imagens como OpenCV e Scikit-Image, o código desenvolvido pode ser utilizado para estudar os efeitos de diversos métodos de manipulação de imagens.

**Desenvolvimento**

O desenvolvimento deste projeto foi realizado em etapas, começando pela implementação de funções básicas para adição de ruído e suavização de imagens, seguindo para a detecção de bordas. Cada etapa foi validada individualmente antes de integrar todas as funcionalidades em um único script. O código foi escrito em Python, utilizando principalmente as bibliotecas numpy, cv2 (OpenCV), matplotlib e skimage.  
O Projeto foi desenvolvido com auxílio de inteligências artificiais como o Chat GPT 3.5.

**Funcionalidades Principais**

* **Adição de Ruídos**: Ruído Gaussiano; Ruído Sal e Pimenta; Ruído Speckle
* **Filtros de Suavização**: Filtro de Média; Filtro de Mediana; Difusão Anisotrópica
* **Detecção de Bordas**: Operador Sobel
* **Interface Gráfica**: Exibição das imagens originais e processadas lado a lado para comparação visual.
* **Salvamento de Imagens**: Salvamento das imagens processadas no disco.

#### **Componentes Principais do Código**

* **Importação de Bibliotecas**: Importa bibliotecas necessárias como skimage, cv2, numpy, e matplotlib.pyplot.
* **Definição de Parâmetros**: Define valores padrão para os parâmetros de ruído, suavização e detecção de bordas.
* **Leitura da Imagem**: Lê a imagem de entrada utilizando a função io.imread.
* **Conversão para Escala de Cinza**: Função que converte a imagem para escala de cinza utilizando cv2.cvtColor.
* **Adição de Ruídos**: Funções que adicionam diferentes tipos de ruído à imagem.
* **Filtros de Suavização**: Funções que aplicam filtros de média, mediana e difusão anisotrópica na imagem.
* **Detecção de Bordas**: Função que aplica o operador Sobel para detecção de bordas.
* **Exibição e Salvamento de Imagens**: Código que exibe as imagens processadas e as salva no disco.

**Cronograma de Desenvolvimento**

Semana 1: Pesquisa e planejamento do projeto, definição de requisitos e estudo das bibliotecas de processamento de imagens.

Semana 2: Implementação das funções de adição de ruídos (Gaussiano, Sal e Pimenta, Speckle) e início da interface que irá se aprimorando conforme o código se desenvolve.

Semana 3: Implementação das funções de filtros de suavização (Média, Mediana, Difusão Anisotrópica).

Semana 4: Implementação da função de detecção de bordas (Operador Sobel).

Semana 5: Aprimoramento da interface gráfica para exibição das imagens.

Semana 6: Testes, depuração e ajustes finais no código e Documentação do código e elaboração do relatório de desenvolvimento.

**Modo de uso**

AAAAAAAAAA

**Conclusão**

Este projeto apresentou um estudo aprofundado sobre a aplicação de diversos filtros e ruídos em imagens, proporcionando uma ferramenta educativa e prática para análise de métodos de processamento de imagens. Através da implementação de funções bem definidas e de fácil utilização, os usuários podem explorar os efeitos de diferentes técnicas e aplicar essas metodologias em seus próprios projetos. Além disso, a estrutura modular do código permite a fácil extensão e personalização conforme novas necessidades surgem.

O tempo estimado para a finalização de todas as etapas do projeto é de **X** horas.

Link do vídeo do programa sendo executado: *poggers.youtube.com.br.poggers*