

Universidade Católica Dom Bosco

Engenharia de Computação

Luan Henrique Santos Miranda
João Henrique Schweitzer Rezende

Convolução em 2D

Campo Grande – MS

2024

Luan Henrique Santos Miranda
João Henrique Schweitzer Rezende

Convolução em 2D

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 – Convolução

Introdução

Este projeto envolve o desenvolvimento de uma aplicação gráfica para aplicar filtros de convolução em imagens utilizando bibliotecas como NumPy, OpenCV e Tkinter. O objetivo é fornecer uma ferramenta interativa que permita a visualização e modificação de imagens através de filtros predefinidos e personalizados.

Desenvolvimento

O Projeto foi desenvolvido com auxílio de inteligências artificiais como o Chat GPT 3.5. Inicialmente no projeto foi pego como base de estudo códigos fornecidos pelo professor regente *Hemerson Pistori*. A aplicação foi desenvolvida em Python, aproveitando o poder das bibliotecas mencionadas para manipulação de imagens e criação de interfaces gráficas. A convolução é uma operação essencial em processamento de imagens, utilizada para aplicar diversos tipos de filtros, como suavização, detecção de bordas, nitidez e efeitos especiais.

Funcionalidades Principais

Carregar Imagem: Permite ao usuário selecionar uma imagem do sistema de arquivos para ser processada.

Aplicar Filtro: Oferece uma seleção de filtros predefinidos que podem ser aplicados à imagem carregada.

Filtro Personalizado: Permite ao usuário definir manualmente a matriz de um filtro personalizado e aplicá-lo à imagem.

Salvar Imagem: Permite salvar a imagem filtrada no sistema de arquivos.

Componentes Principais do Código

- `def convolution(image, kernel):` Função que aplica a operação de convolução a uma imagem usando um kernel especificado.
- `def load_image():` Função que abre uma janela de diálogo para carregar uma imagem e exibi-la.
- `def save_image(image):` Função que salva a imagem filtrada após o processo de convolução.
- `def apply_filter():` Função que aplica o filtro pré estabelecido selecionado à imagem carregada.

- `def confirm_dimension()`: Função que confirma as dimensões do filtro personalizado inseridas pelo usuário e cria os campos para entrada dos valores.
- `def create_custom_filter_window()`: Função que cria uma janela para o usuário inserir a matriz de um filtro personalizado.
- `def create_filter_matrix_window(num_rows, num_cols)`: Função que cria a interface para entrada dos valores do filtro personalizado.
- `def apply_filter_custom()`: Função que aplica um filtro personalizado definido pelo usuário à imagem.

Cronograma de Desenvolvimento

Semana 1: Pesquisa e planejamento do projeto, definição de requisitos e idealização das funções do projeto.

Semana 2: Início da implementação das funções base como carregamento de imagens e primeira versão da função *convolution*.

Semana 3: Agora temos em um arquivo secundário para armazenar os filtros pré estabelecidos, e aplicando de maneira mais otimizada, agora é possível salvar a imagem após o processamento de convolução.

Semana 4: Criação da interface com Tkinter, que sempre será atualizada com base nas novas funções do código base.

Semana 5: Adicionado a possibilidade de ser criado Filtros Personalizados com os valores que o usuário desejar, além disso é mostrado o *kernel* tanto na interface quanto no terminal.

Semana 6: Testes e refinamentos finais, documentação do código e preparação do relatório.

Modo de uso

Para utilizar o projeto, siga os seguintes passos abaixo:

Certifique-se de ter o “*conda*” instalado, após isso, dê o seguinte comando no terminal (base) para permitir que rode o script bash desenvolvido para que rode a aplicação da maneira correta:

```
chmod +x script.sh
```

Ainda no terminal o usuário deve dar o seguinte comando, iniciando assim, todo o processo de maneira automática, desde a criação do ambiente conda, instalação das dependências, até o processamento do código:

```
./script.sh
```

Já na execução do programa, ele possui uma interface gráfica intuitiva, sendo assim, de fácil utilização, onde o usuário precisa somente seguir em ordem as etapas, para correto funcionamento do código, sendo primeiramente carregar uma imagem, após isso é necessário fechar a imagem, escolher o filtro que deseja, e clicar em aplicar, isso irá mostrar a imagem alterada, para aplicar outro filtro, é necessário por garantia fechar a imagem. A aplicação tem também uma funcionalidade onde o usuário personaliza um filtro de sua escolha, podendo escolher a quantidade de linhas e colunas que serão necessárias. Tendo também a possibilidade de salvar a imagem.

Carregar Imagem: Clique no botão "Carregar Imagem" e selecione uma imagem do seu sistema de arquivos.

Selecionar Filtro: Use a lista suspensa para selecionar um filtro predefinido ou escolha "Filtro Personalizado" para criar o seu próprio.

Aplicar Filtro: Clique no botão "Aplicar Filtro" para aplicar o filtro selecionado à imagem carregada. Se for um filtro personalizado, insira a matriz do filtro e confirme antes de aplicar.

Salvar Imagem: Clique no botão "Salvar Imagem" para salvar a imagem filtrada no seu sistema de arquivos.

Conclusão

O projeto resultou em uma ferramenta interativa e funcional para aplicar filtros de convolução em imagens. Com uma interface e a flexibilidade de definir filtros personalizados, a aplicação oferece uma ampla gama de possibilidades para manipulação de imagens.

O tempo estimado para a finalização de todas as etapas do projeto é de **20** horas.

Link do vídeo do programa sendo executado: [Projeto Convolução 2d | Visão Computacional | Luan Henrique e João Schweitzer](#)

