Disciplina: Processamento de Imagens de Computação Gráfica
Curso: Ciência da Computação
Docente: Profª Ma. Layse Souza

Descrição do projeto

A cidade de Véridia deseja criar um sistema para analisar, transformar e realçar imagens digitais em diferentes contextos, como educação, saúde e indústria. A prefeitura contratou vocês para desenvolver um Sistema de Processamento de Imagens, permitindo que operadores e administradores processem imagens, apliquem filtros, transformações e análises de padrões, mantendo registro de resultados e relatórios de processamento. O sistema deve ser implementado em Python, utilizando bibliotecas como numpy, cv2, PIL, skimage e matplotlib.

Os objetivos desse sistema são:

- Centralizar o processamento de imagens em uma única plataforma;
- Facilitar a aplicação de filtros e transformações de forma controlada;
- Garantir consistência e integridade dos resultados;
- Permitir análise detalhada de padrões e componentes de imagens;
- Incentivar pesquisa e aplicação de novas técnicas em processamento de imagens.

Nesta etapa, os grupos deverão implementar semanalmente as etapas de um sistema modular e inteligente capaz de analisar, transformar, comparar e gerar relatórios sobre imagens digitais.

O sistema deve ser desenvolvido em Python, utilizando bibliotecas como numpy, opencv (cv2), PIL, matplotlib, scikit-image, scikit-learn, reportlab e, opcionalmente, tensorflow ou streamlit para interface interativa.

Cada grupo terá tarefas específicas e complementares, com atividades principais múltiplas e desafios avançados que visam integrar ciência de dados, visão computacional e automação de relatórios.

Plano de Implementação - Entregas

SEMANAS (8,0)	OBJETIVOS
Semana 01 (14/10) - Estrutura e planejamento do módulo (1,0)	- Criar a base do projeto em Python no Google Colab ou ambiente local. - Definir as funções principais e o fluxo de execução do módulo. - Iniciar a configuração do repositório no GitHub e inserir o README inicial.
Semana 02 (28/10) - Implementação das funcionalidades principais (1,5)	- Desenvolver as funções centrais definidas na Unidade I. - Testar o funcionamento com diferentes imagens. - Registrar resultados iniciais e atualizar o repositório.
Semana 03 (04/11) - Aprimoramento e análise dos resultados (2,0)	- Realizar novos testes com outras imagens. - Corrigir falhas de execução e aprimorar os resultados visuais. - Inserir prints e tabelas de comparação no repositório.
Semana 04 (11/11) - Documentação, análise dos resultados e vídeo de demonstração (2,0)	- Elaborar a documentação parcial em formato .pdf, apresentando objetivos, metodologia, imagens usadas e resultados obtidos. - Adicionar descrição técnica no README.md e no arquivo final. - Produzir um vídeo curto (máx. 5 min) demonstrando o funcionamento do módulo e os resultados alcançados. - Publicar o vídeo na pasta /demo e finalizar o repositório com commits organizados.
Semana 05 (18/11) - Revisão e entrega final (1,5)	- Revisar todo o código, limpar comentários, ajustar nomes de arquivos e garantir a execução correta do projeto. - Entregar o link final do repositório no Google Classroom.

Tarefas dos grupos
● ***Grupo 1 - Ajuste de brilho e contraste*** 1. Implementar o que foi proposto na Unidade I. 2. Implementar normalização de brilho global com base no histograma YCrCb. 3. Criar função de equalização adaptativa (cv2.createCLAHE). 4. Automatizar ajuste de contraste por região da imagem. 5. Desenvolver um "mapa de contraste" que mostre áreas com excesso ou falta de brilho. 6. Registrar resultados em arquivo .CSV ou imagem de saída. 7. Documentar. 8. Elaborar um artigo científico demonstrando todo o processo realizado.
● ***Grupo 2 - Conversão de imagens para escala de cinza*** 1. Implementar o que foi proposto na Unidade I. 2. Converter imagens coloridas para cinza após detecção de ruído. 3. Aplicar diferentes filtros (Gaussian, Bilateral, Median) e comparar resultados. 4. Criar função para medir nitidez (foco) com Laplacian. 5. Construir um avaliador de qualidade de imagem baseado em ruído e foco. 6. Armazenar resultados em tabela comparativa ou imagem processada. 7. Documentar. 8. Elaborar um artigo científico demonstrando todo o processo realizado.
● ***Grupo 3 - Rotação e redimensionamento*** 1. Implementar o que foi proposto na Unidade I. 2. Implementar redimensionamento automático com preservação da proporção. 3. Criar função de rotação manual e automática (detectando ângulo com HoughLines). 4. Realizar recorte automático com base em ROI detectada. 5. Desenvolver uma função "auto-align" que corrige a orientação de imagens inclinadas. 6. Implementar uma ferramenta que permita gerar miniaturas automaticamente em diferentes tamanho para visualização rápida 7. Documentar. 8. Elaborar um artigo científico demonstrando todo o processo realizado.
● ***Grupo 4 - Filtros de suavização e borda*** 1. Implementar o que foi proposto na Unidade I. 2. Implementar detecção de bordas simples (gradiente de intensidade). 3. Comparar resultados obtidos entre os diferentes tipos de filtros. 4. Calcular o tempo de processamento e impacto na qualidade final. 5. Criar visualização lado a lado das imagens original e processada. 6. Desenvolver função que ajuste automaticamente o nível de suavização. 7. Documentar. 8. Elaborar um artigo científico demonstrando todo o processo realizado.

- ***Grupo 5 - Transformações lineares e não-lineares***

1. Implementar o que foi proposto na Unidade I.
2. Aplicar transformação linear (contraste, brilho e inversão) e transformação não-linear (logarítmica e gama).
3. Comparar resultados e histogramas entre os métodos aplicados.
4. Criar função de ajuste dinâmico de parâmetros.
5. Medir o impacto das transformações de luminosidade geral da imagem.
6. Desenvolver um gráfico de evolução do contraste entre métodos.
7. Documentar.
8. Elaborar um artigo científico demonstrando todo o processo realizado.

- ***Grupo 6 - Realce no domínio espacial***

1. Implementar o que foi proposto na Unidade I.
2. Aplicar técnicas de realce de nitidez no domínio espacial.
3. Implementar equalização local de contraste (CLAHE).
4. Comparar resultados de realce global e local, e medir nitidez e variação de intensidade antes e depois do realce.
5. Criar visualização comparativa entre diferentes parâmetros de realce.
6. Desenvolver função híbrida combinando suavização e realce local.
7. Documentar.
8. Elaborar um artigo científico demonstrando todo o processo realizado.

- ***Grupo 7 - Manipulação de canais de cor***

1. Implementar o que foi proposto na Unidade I.
2. Separar canais RGB, HSV e LAB.
3. Calcular estatísticas (média, desvio, skewness) por canal.
4. Aplicar equalização apenas em um canal específico e observar os efeitos.
5. Criar um comparador de canais com visualização dos histogramas individuais.
6. Avaliar efeitos visuais das manipulações de cor.
7. Documentar.
8. Elaborar um artigo científico demonstrando todo o processo realizado.

- ***Grupo 8 - Análise de vizinhança e detecção de componentes***

1. Implementar o que foi proposto na Unidade I.
2. Implementar análise de vizinhança 4 e 8 em imagens binárias.
3. Calcular conectividade e distância entre pixels, e identificar regiões e componentes simples.
4. Gerar matriz de conectividade e visualização colorida das regiões.
5. Criar função que conte o número de componentes detectados.
6. Comparar tempos e precisões para diferentes métodos de vizinhança.
7. Documentar.
8. Elaborar um artigo científico demonstrando todo o processo realizado.

ENTREGA NO CLASSROOM (Google Sala de Aula):

- **Cada grupo será avaliado semanalmente**
- **Todos os integrantes devem ser adicionados como colaboradores do repositório**
- **O link do repositório deverá ser postado no Google Classroom, conforme as datas de entrega**
- **Alterações e commits serão utilizados como parte da avaliação de participação individual**
- **As versões entregues devem estar funcionais, documentadas e acompanhadas do vídeo demonstrativo**
- **Cada grupo deverá criar um repositório público no GitHub com o nome: Processamento de Imagens_E01_GrupoX (substituindo o "X" pelo número do grupo)**
    - **O repositório deverá conter, obrigatoriamente, as seguintes pastas e arquivos:**
        - **/src → códigos em Python (.ipynb ou .py)**
        - **/imagens → conjunto de imagens utilizadas no projeto (.png, .jpg, .jpeg)**
        - **/docs → documentação parcial (entregas semanais) e final em formato .PDF**
        - **README.md → arquivo explicativo com:**
            - **Objetivo do módulo desenvolvido**
            - **Bibliotecas utilizadas**
            - **Instruções de execução**
            - **Responsabilidades de cada integrante**
            - **Prints ou exemplos de saída**
        - **/demo → vídeo curto (máximo de 5 minutos) demonstrando o funcionamento básico do sistema**
            - **O vídeo deve ter formato .mp4**
            - **Pode mostrar apenas a execução e os resultados obtidos, sem necessidade de narração**