**Resumo do Projeto: Processamento de Imagens com Python**

**Objetivo do Projeto**

O objetivo principal deste projeto foi desenvolver um processo automatizado para realizar o pré-processamento de imagens. Este processo envolveu etapas como redimensionamento, ajuste de cor, correção de brilho e contraste, rotação e orientação das imagens. O foco foi melhorar a qualidade das imagens e torná-las adequadas para análise posterior, como o reconhecimento óptico de caracteres (OCR).

**Etapas do Projeto**

1. **Planejamento e Escolha das Ferramentas** A primeira etapa foi planejar as ferramentas e tecnologias que seriam usadas no projeto. Para isso, escolhi a linguagem Python devido à sua robustez e à variedade de bibliotecas para processamento de imagens. As bibliotecas utilizadas foram:
   * **OpenCV**: Para o processamento de imagens (ajustes de brilho, contraste, redimensionamento, etc.).
   * **Pytesseract**: Para realizar OCR nas imagens.
   * **Pillow**: Para realizar ajustes finos, como rotação e correção de cor.
2. **Aquisição das Imagens** As imagens utilizadas no projeto foram coletadas de uma pasta chamada original\_images/. Essas imagens estavam em formatos variados e, em alguns casos, apresentavam problemas como orientação incorreta ou baixa qualidade.
3. **Desenvolvimento do Código** O código Python foi desenvolvido para realizar as seguintes etapas de pré-processamento:
   * **Redimensionamento**: Ajuste das imagens para um tamanho padrão, garantindo consistência.
   * **Correção de Cor**: Ajuste no balanço de cor para melhorar a visibilidade.
   * **Ajustes de Brilho e Contraste**: Melhora da iluminação e definição das imagens.
   * **Rotação e Orientação**: Correção da orientação das imagens para garantir que os textos ou elementos nas imagens ficassem na posição correta.

Durante esse processo, foram gerados arquivos de saída na pasta processed\_images/, com nomes indicando a etapa de processamento realizada (ex.: 1\_redimensionado\_, 2\_cinza\_, etc.).

1. **Testes** Para garantir que o código estava funcionando corretamente, diversas imagens foram processadas e verificadas. Os resultados foram analisados visualmente, verificando se as imagens estavam mais claras, nítidas e corretamente orientadas. Em alguns casos, o código foi ajustado para melhorar o desempenho em imagens específicas.
2. **Implementação do OCR** Após o pré-processamento, as imagens passaram por um processo de OCR utilizando a ferramenta **Tesseract OCR**, via a biblioteca pytesseract. O objetivo era verificar se o código conseguia realizar a extração de texto das imagens processadas de maneira eficiente. As imagens corrigidas foram enviadas para esse processamento, e os resultados foram comparados.

**Resultados Obtidos**

O projeto alcançou com sucesso seu objetivo de processar as imagens, realizando ajustes na qualidade visual e na orientação das mesmas. Algumas imagens que antes estavam viradas ou com cores distorcidas foram corrigidas. O processo de OCR também funcionou bem nas imagens após o pré-processamento, com boa taxa de reconhecimento de caracteres.

As imagens finais foram salvas na pasta processed\_images/ e podem ser usadas para análises ou outras aplicações, como extração de texto ou análise de dados visuais.

**Conclusão**

Este projeto demonstrou a eficácia de usar Python e suas bibliotecas de processamento de imagens para automatizar e melhorar a qualidade das imagens antes de realizar OCR ou outras análises. O código foi estruturado de forma modular, facilitando futuras melhorias e adições de novas etapas de pré-processamento. O uso de GitHub também garantiu um controle de versão eficiente e facilitou o trabalho colaborativo. Além disso, a realização de testes em diferentes ambientes mostrou que o projeto é robusto e pode ser executado em diversas máquinas.