UNIVERSIDADE TIRADENTES – UNIT

CIÊNCIA DA COMPUTAÇÃO

Ian Lucas Costa Santana

Antonny Silva Cruz

Pedro Luiz Pinho Gois

Gabriel Schettino Santos

Igor Gabriel Souza Lima

SISTEMA DE MÓDULOS PARA ROTAÇÃO E REDIMENSIONAMENTO DE IMAGENS DIGITAIS

Aracaju – SE

2025

Ian Lucas Costa Santana
Antonny Silva Cruz
Pedro Luiz Pinho Gois
Gabriel Schettino Santos
Igor Gabriel Souza Lima

SISTEMA DE MÓDULOS PARA ROTAÇÃO E REDIMENSIONAMENTO DE IMAGENS DIGITAIS

ATIVIDADE sobre rotação e redimensionamento de imagens digitais como requisito parcial da avaliação da disciplina processamento de imagens ministrada pela Prof. Layse Santos Souza, no 2° semestre de 2025.

Aracaju - SE
2025

Sumário

1. Objetivos

1.1 Objetivo Geral

Criar módulos em Python que permitam o redimensionamento e a rotação de imagens digitais, preservando a proporção e a qualidade final.

1.2 Objetivos Específicos

- Upload individual ou múltiplo de imagens
- Pré-visualização antes do processamento
- Redimensionamento sem distorção
- Rotação em ângulos fixos e livres
- Controle da qualidade mínima (ex.: JPEG ≥ 80%)
- Comparação entre imagem original e processada
- Registro em histórico para auditoria
- Suporte a múltiplos usuários simultâneos

2. Metodologia

A metodologia utilizada para o processo de implementação do código e do projeto foi linear e sequencial, onde cada fase (determinando o tema do projeto; levantamento de objetivos, requisitos, funcionalidades e fluxo do programa; implementação; testes; implantação e manutenção) foi realizada a cada semana proposta do percurso do projeto.

3. Imagens usadas

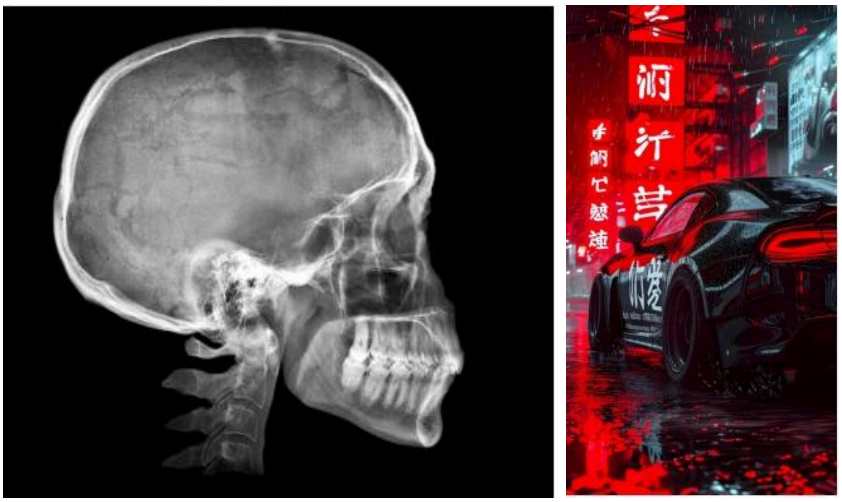

Figura 1 e 2 - Raio-x e foto de carro

4. Resultados obtidos

4.1 Primeira Semana

Criação da base do projeto em Python dentro do google colab para começa a implementar os códigos das funções principais definidas na primeira parte do projeto realizado anteriormente, como também, a criação do repositório Processamento-de-Imagens_E01_Grupo3.

4.2 Segunda Semana

Desenvolvimento das funções Centrais, como por exemplo:

```python
def upload_imagens_colab():
    """Upload de imagens no Colab e retorna lista de (caminho, imagem)."""
    uploaded = files.upload()
    imgs = []
    for nome in uploaded.keys():
        caminho = nome
        img = cv2.imread(caminho)
        if img is not None:
            imgs.append((caminho, img))
        else:
            print(f"Erro ao carregar {caminho}")
    return imgs

def validar_imagem(img):
    return img is not None and img.shape[0] > 10 and img.shape[1]
```

Figura 3 - Funções upload e validar_imagem

Testes do funcionamento com diferentes imagens e registros dos resultados iniciais e atualização do repositório através de commits no GitHub.

4.3 Terceira Semana

Realização de testes com outras imagens, como por exemplo:

```python
def recorte_automatico(img, padding=5):
    """
    Detecta a maior ROI por contornos após threshold adaptativo e retorna o recorte.
    Se não encontrar contornos relevantes, retorna a imagem original.
    """
    if img is None:
        return img
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) if len(img.shape) == 3 else img.copy()
    blur = cv2.GaussianBlur(gray, (5,5), 0)
    thresh = cv2.adaptiveThreshold(blur, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                   cv2.THRESH_BINARY_INV, 11, 2)
    kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (5,5))
    clean = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE, kernel, iterations=1)

    contornos, _ = cv2.findContours(clean, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    if not contornos:
        return img

    maior = max(contornos, key=cv2.contourArea)
    x, y, w, h = cv2.boundingRect(maior)

    x0 = max(x - padding, 0)
    y0 = max(y - padding, 0)
    x1 = min(x + w + padding, img.shape[1])
    y1 = min(y + h + padding, img.shape[0])

    return img[y0:y1, x0:x1]

def auto_align(img, debug=False):
    """
    Pipeline automático: recorte por ROI -> detecção de inclinação -> correção.
    Retorna imagem recortada e alinhada.
    """
    if img is None:
        return img
    rec = recorte_automatico(img)
    rot = rotacao_automatica(rec, debug=debug)
```

Figura 4 - Funções recorte automático e auto-align

Correção de falhas de execução e aprimoramento dos resultados visuais e inserção dos prints e tabelas de comparação no repositório.

4.4 Quarta Semana

Elaboração do documento parcial e envio no formato .PDF apresentando os objetivos, metodologias e imagens usadas.

Produção do vídeo para apresentação do projeto demonstrando os módulos de funcionamento e objetivos alcançados e publicação dos resultados na Pasta/DEMO.

Segue os seguintes resultados dos códigos obtidos:

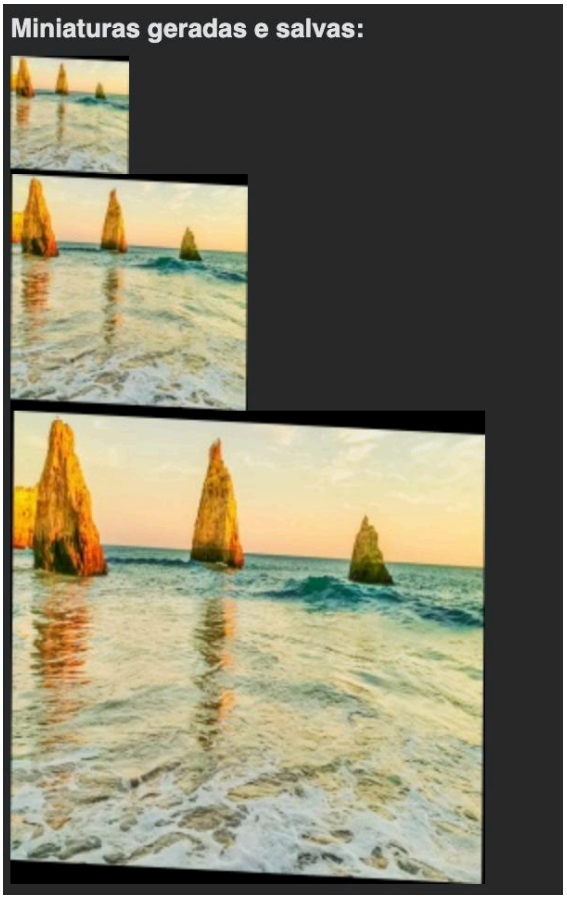

Figura 5 - Resultado de miniaturas

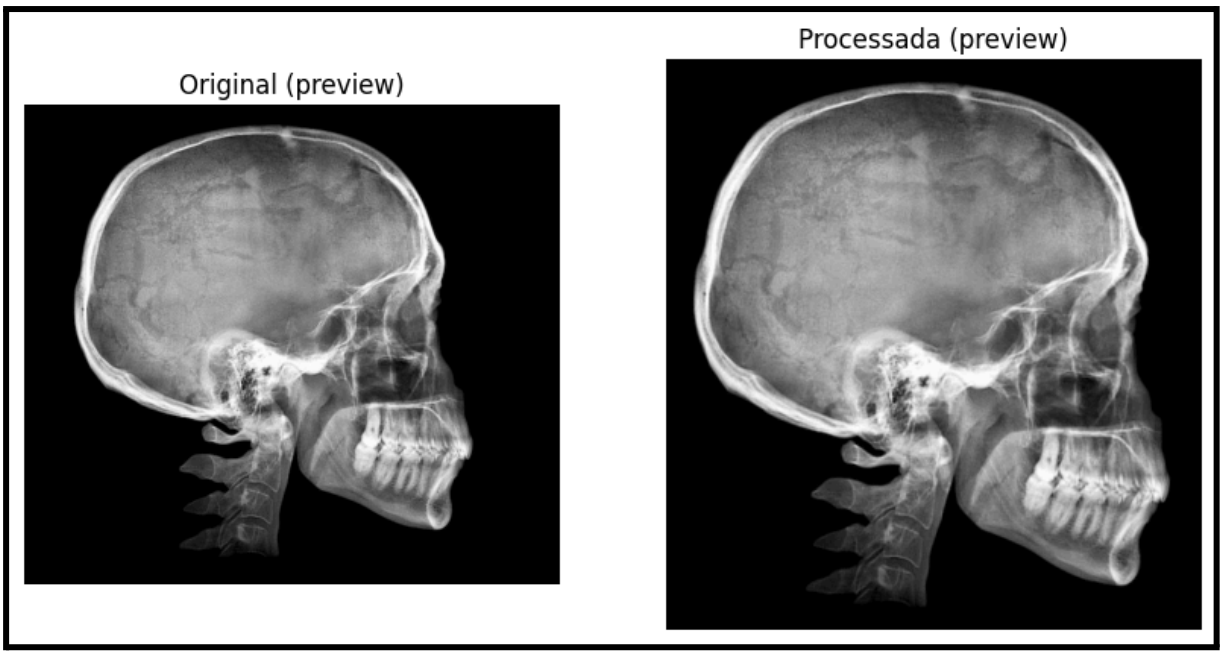

Figura 6 - Resultado da função do auto-align

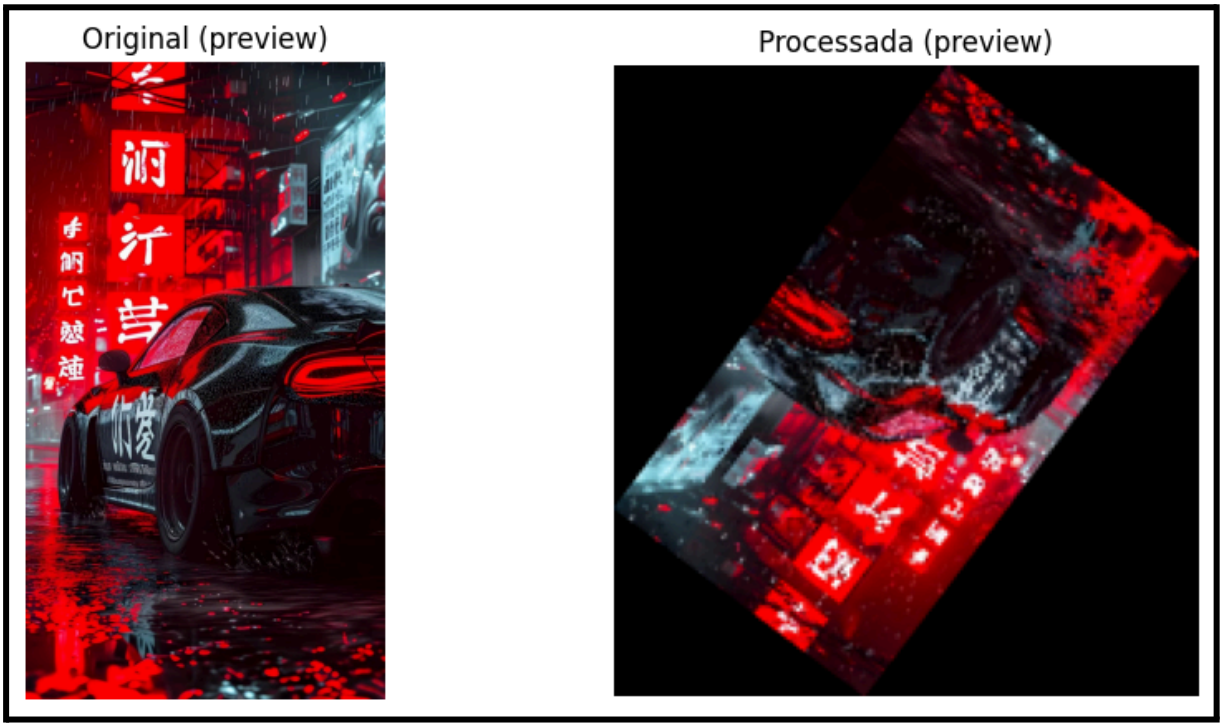

Figura 7 - Resultado de rotação de imagem

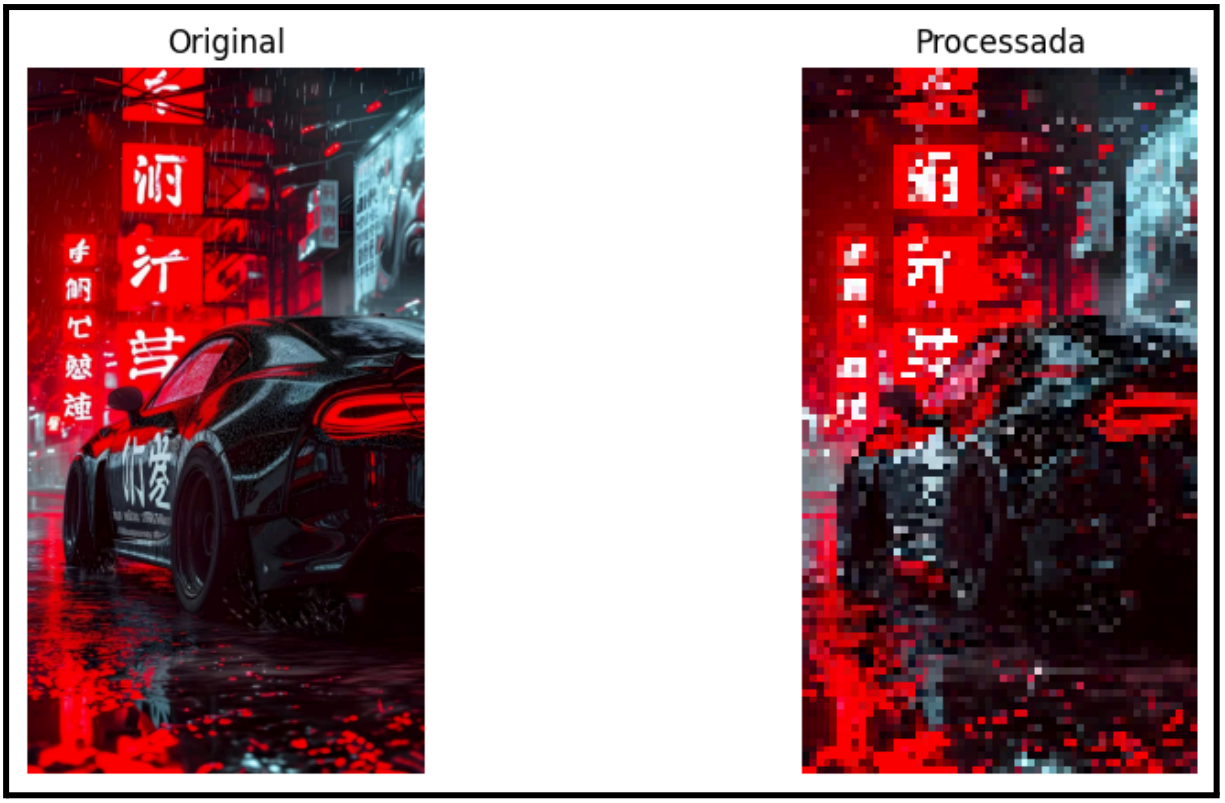

Figura 8 - Resultado de largura

5. Aplicações Práticas

O sistema de módulos para rotação e redimensionamento de imagens digitais possui diversas aplicações práticas em contextos reais de processamento e análise de imagens. A seguir, são apresentadas algumas áreas em que essas funcionalidades são amplamente utilizadas:

Sistemas de Reconhecimento Facial:
O redimensionamento é essencial para padronizar o tamanho das imagens antes do treinamento de modelos de reconhecimento facial, garantindo que todas as faces tenham a mesma escala. A rotação, por sua vez, ajuda a corrigir inclinações no enquadramento do rosto, melhorando a precisão do reconhecimento em sistemas de segurança e autenticação biométrica.

Arquivamento e Digitalização de Documentos:

Em sistemas que digitalizam documentos, o redimensionamento permite reduzir o tamanho dos arquivos sem perda significativa de qualidade, otimizando o armazenamento. Já a rotação é usada para corrigir documentos escaneados que foram capturados fora de alinhamento, garantindo uma visualização correta.

Plataformas de E-commerce e Mídias Sociais:

A manipulação automática de imagens (como redimensionar e girar) é usada para padronizar fotos de produtos ou postagens, mantendo a estética e proporções uniformes nas interfaces.

Sistemas Médicos e Radiológicos:

No contexto de imagens médicas (como radiografias e tomografias), a rotação e o redimensionamento são usados para alinhar imagens e ajustá-las para análise comparativa, preservando a qualidade e a proporção das regiões de interesse.

Essas aplicações demonstram a relevância dos módulos desenvolvidos, que podem ser facilmente adaptados e integrados em diferentes contextos tecnológicos e científicos.

6. Conclusão

O desenvolvimento dos módulos para rotação e redimensionamento de imagens digitais proporcionou uma compreensão prática sobre as etapas de manipulação e processamento de imagens. Além de alcançar os objetivos propostos, o grupo percebeu a importância dessas operações como base para aplicações mais complexas, como sistemas de reconhecimento facial, análise médica e automação de imagens digitais. O projeto demonstrou, na prática, a integração entre teoria e implementação em Python, utilizando bibliotecas como *OpenCV*, *Pillow* e *scikit-image*.