

# Processamento Digital de Imagens

Professor: Dr. Franklin Cesar Flores

Acadêmico: Munif Gebara Junior R.A.:pg 5575-2

## Introdução

O Processamento Digital de Imagens (PDI) é uma área central da visão computacional que visa o aprimoramento, análise e extração de informações visuais a partir de imagens digitais. Entre as diversas transformações possíveis, operações baseadas em intensidade e cor são amplamente utilizadas para melhorar o contraste, segmentar regiões de interesse ou preparar imagens para análises subsequentes. Neste contexto, a representação de imagens no espaço de cores HSV (Hue, Saturation, Value) torna-se particularmente útil, pois permite isolar o componente de brilho (Value) dos demais atributos cromáticos (matiz e saturação), possibilitando intervenções mais direcionadas e perceptualmente relevantes.

## Modelo HSV

O modelo HSV é uma transformação do espaço RGB que separa informações de cor e intensidade de forma mais coerente com a percepção humana. Neste modelo: Hue (H) representa a tonalidade da cor (0–360 graus), como vermelho, verde, azul etc.; Saturation (S) indica o grau de pureza da cor (0 = cinza, 100% = cor pura); Value (V) corresponde à intensidade ou brilho da cor, variando entre 0 (preto) e 255 (branco). Esta decomposição permite, por exemplo, manipular o brilho de uma imagem mantendo as cores originais inalteradas, ou converter a imagem para tons de cinza atuando apenas sobre os componentes H e S.

## Passagem de HSV para Tons de Cinza

Uma conversão efetiva de uma imagem HSV para uma aparência monocromática (preto e branco ou tons de cinza) pode ser realizada zerando os componentes H e S, mantendo o canal V intacto. Isso resulta em uma imagem cuja aparência depende exclusivamente da intensidade luminosa, já que a matiz e a saturação — atributos cromáticos — são anulados. Esta técnica é útil para analisar a distribuição de brilho na imagem sem a interferência de informações de cor.

## Operações de Limite (Thresholding)

A operação de limiarização (ou thresholding) consiste em segmentar uma imagem com base em um valor de intensidade: todos os pixels com intensidade inferior a um limiar predefinido são classificados como "baixos", enquanto os superiores são classificados como "altos". Este tipo de operação é comum em binarização e realce seletivo de regiões com determinadas faixas de brilho.

## Limiar de Otsu

O método de Otsu (1979) é uma técnica não supervisionada de seleção de limiar ótimo baseada na minimização da variância intra-classe ou, equivalentemente, maximização da variância entre classes. O algoritmo analisa o histograma da imagem para encontrar o valor de limiar  $T^*$  que melhor separa os pixels em dois grupos distintos (geralmente, fundo e objeto), sem conhecimento prévio do conteúdo da imagem.

## Contraste e Normalização

Contraste refere-se à diferença visual entre tons claros e escuros em uma imagem. Baixo contraste resulta em imagens "lavadas" ou escuras, enquanto alto contraste enfatiza bordas e formas. A normalização é uma técnica clássica para maximizar o contraste dinâmico de uma imagem, expandindo a faixa de intensidades para um intervalo padrão, geralmente  $[0, 255]$ . Esse procedimento permite que diferenças sutis de intensidade se tornem mais perceptíveis, melhorando a visualização e o desempenho de algoritmos de segmentação.

## Ajuste de Gama

A correção gama é uma transformação não linear utilizada para ajustar a luminosidade de uma imagem de acordo com a resposta perceptual humana. O  $\gamma > 1$  realça regiões claras da imagem. Útil para compensar imagens subexpostas ou para realçar brilhos sutis. O  $0 < \gamma < 1$  realça regiões escuras da imagem. Escurece tons médios e altos. Indicado para imagens superexpostas ou para enfatizar sombras.

A escolha de diferentes valores de gama para diferentes regiões da imagem (por exemplo, com base em um limiar como o de Otsu) permite um controle localizado do contraste, maximizando a percepção em múltiplas faixas dinâmicas simultaneamente.

Neste trabalho, tais fundamentos teóricos foram integrados de forma modular para compor um conjunto de experimentos práticos aplicados a imagens reais. As operações foram implementadas em Python utilizando bibliotecas como NumPy, OpenCV e Matplotlib, permitindo manipulações diretas no espaço HSV. A partir disso, foram desenvolvidas abordagens específicas de realce de brilho, conversão seletiva para tons de cinza, limiarização automática via Otsu, normalização dinâmica e correção gama adaptativa, com aplicação diferenciada em regiões escuras e claras da imagem. Esses recursos foram combinados em pipelines flexíveis que possibilitam a análise comparativa entre métodos, visando a compreensão dos efeitos visuais e computacionais decorrentes de cada técnica.

A solução faz a leitura de todas as imagens na pasta **input** e manipula e escreve as imagens na pasta **output**. As imagens utilizadas antes e depois da normalização tradicional estão na Figura 1. As imagens bar e igreja são iguais na entrada e na saída, já a imagem show apresenta diferença. Isso ocorre porque o V máximo e mínimo é 0 e 255 nas imagens bar e igreja e 10 e 177 na imagem show.



Figura 1 - Imagens Utilizadas antes e depois da normalização tradicional em V

A normalização tradicional é aplicada no espaço HSV alterando apenas o V como a função python no Código 1, em destaque a função de normalização.

A primeira variação da normalização é feita através de um threshold manual mas com saídas V dos pixels normalizados entre 0 - 255. As imagens estão na Figura 2. Embora para as imagens bar e igreja os resultados tenham sido interessantes, a imagem show ficou ruim.

# Projeto de Processamento Digital de Imagens (PDI)

Este projeto realiza operações de processamento digital de imagens utilizando Python 3, com as bibliotecas OpenCV, NumPy e Matplotlib. Ele inclui funções para:

- Carregar e salvar imagens
- Conversão para HSV
- Normalização do canal V (brilho)
- Cálculo de limiar com o método de Otsu
- Correção gama com limiar adaptativo
- Transformação HSV → escala de cinza
- Visualização lado a lado

## Estrutura de pastas

projeto-pdi/

```
|— input/      # Imagens de entrada (.jpg)
|— output/     # Imagens processadas
|— utils_pdi.py # Funções auxiliares
|— main.py     # Script principal de execução
|— requirements.txt
|— README.md
```

## Requisitos

- Python 3.8+
- NumPy
- OpenCV
- Matplotlib

Instale com:

```
pip install -r requirements.txt
```

## Como executar






Execute o script principal:

```
python main.py
```

O script irá:

1. Processar todas as imagens **.jpg** na pasta **input/**
2. Aplicar transformação de brilho, correção gama ou outra operação conforme configurado
3. Salvar os resultados na pasta **output/**
4. Exibir visualização comparativa (original vs. processada)

## Funcionalidades implementadas

-  Normalização de brilho (canal V)
-  Correção gama adaptativa (dupla)
-  Aplicação de limiar de Otsu
-  Conversão para preto e branco via HSV
-  Visualização com `matplotlib`

## Exemplos de uso

```
from utils_pdi import normalizar_v_com_limite, corrigir_gama_duplo
img_hsv_norm = normalizar_v_com_limite(img_hsv, limite_v=80)
img_hsv_corr = corrigir_gama_duplo(img_hsv, gama_baixo=2.0, gama_alto=0.8)
```

Todas as Funções:

Função	Parâmetros	Descrição
<code>carregar_imagem</code>	<code>caminho</code>	Carrega uma imagem BGR a partir do caminho informado.
<code>converter_para_hsv</code>	<code>imagem_bgr</code>	Converte uma imagem BGR para HSV (float32).
<code>salvar_imagem</code>	<code>imagem_bgr</code> , <code>caminho_saida</code>	Salva uma imagem BGR no caminho especificado.
<code>mostrar_imagem</code>	<code>imagem_bgr</code> , <code>titulo='Imagem'</code>	Exibe uma imagem BGR com Matplotlib.
<code>mostrar_imagens_lado_a_lado</code>	<code>img1_bgr</code> , <code>img2_bgr</code> , <code>titulo1</code> , <code>titulo2</code>	Mostra duas imagens lado a lado.
<code>normalizar_v</code>	<code>hsv_img</code>	Normaliza todo o canal V da imagem HSV.
<code>normalizar_v_com_limite</code>	<code>hsv_img</code> , <code>limite_v</code>	Normaliza o V apenas onde $V < \text{limite\_v}$ .
<code>normalizar_v_com_limite_limitado</code>	<code>hsv_img</code> , <code>limite_v</code>	Normaliza $V < \text{limite\_v}$ com reescala limitada.
<code>limiar_otsu</code>	<code>v</code>	Calcula o limiar ótimo com o método de Otsu para o canal V.
<code>normalizar_v_com_limite_limitado_otsu</code>	<code>hsv_img</code>	Aplica normalização limitada onde $V < \text{limiar de Otsu}$ .
<code>corrigir_gama_duplo</code>	<code>hsv_img</code> , <code>gama_baixo</code> , <code>gama_alto</code>	Aplica correção gama dupla no canal V com base no limiar de Otsu.
<code>hsv_para_pb</code>	<code>hsv_img</code>	Remove cor (zera H e S), mantendo o brilho V para efeito em tons de cinza.