

# Segmentação de Imagens de Jardins Utilizando Texturas

Jennifer M. de P. Goberski<sup>1</sup>, Eduardo Todt<sup>1</sup>

<sup>1</sup>Departamento de Informática – Universidade Federal do Paraná (UFPR)

jennifer.goberski@ufpr.br, todt@inf.ufpr.br

**Resumo.** Este trabalho propõe a segmentação de imagens baseada em textura, utilizando um conjunto de filtros aplicados em diferentes escalas. Foi selecionado um conjunto de 16 imagens do tipo jardim florido para análise. Para caracterizar a textura, empregaram-se filtros com orientações horizontal, vertical, 45°, 135° e circular. Esses filtros foram aplicados em três escalas distintas, obtidas por meio da aplicação sucessiva de um filtro gaussiano seguido da redução da imagem à metade de suas dimensões.

Cada região da imagem foi descrita por vetores compostos pela média das respostas dos filtros em janelas locais. Posteriormente, esses vetores foram agrupados com base na similaridade de textura, utilizando a distância euclidiana em espaço N-dimensional para formar grupos de regiões semelhantes. A imagem final apresenta a categorização por textura, destacando visualmente os agrupamentos formados.

A metodologia foi implementada em Python com uso da biblioteca OpenCV, sem o emprego de deep learning, focando apenas em técnicas clássicas de processamento de imagem e análise de padrões.

## 1. Descrição das Imagens Analisadas

Este trabalho utiliza um conjunto de 16 imagens de jardins floridos, obtidas em domínio público. As imagens foram escolhidas por apresentarem diversidade de texturas e cores entre os elementos naturais, como flores, folhas, caules, pedras e objetos decorativos presentes nos cenários. A variedade visual encontrada nessas cenas permite avaliar a capacidade dos filtros de textura em diferenciar regiões com padrões distintos, contribuindo para uma segmentação mais coerente com a estrutura e a aparência dos elementos representados.

## 2. Ambiente de Desenvolvimento

O ambiente de desenvolvimento adotado neste trabalho é baseado na linguagem *Python*, utilizando a biblioteca *OpenCV* para a aplicação de técnicas clássicas de processamento de imagem e análise de padrões. Essa abordagem foi empregada tanto na construção dos filtros de textura quanto na manipulação multiescala das imagens analisadas. O desenvolvimento foi realizado por meio de *Jupyter Notebooks*, com o uso da plataforma *Google Colab*, que oferece uma interface acessível para experimentação, além de facilitar o compartilhamento do código e dos resultados obtidos em cada etapa do processo.

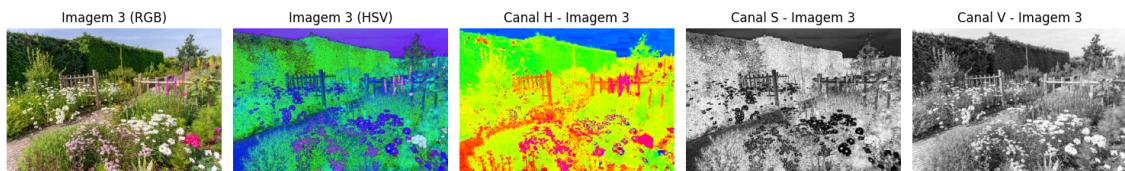
### 3. Manipulação das Imagens

As 16 imagens utilizadas no experimento foram previamente organizadas em uma pasta no *Google Drive* e, em seguida, carregadas no ambiente *Google Colab* para processamento. A Figura 1 apresenta uma imagem de exemplo da base utilizada.



**Figure 1. Exemplos de imagens utilizadas**

Após o carregamento, as imagens no formato RGB foram convertidas para o espaço de cor HSV (*Hue, Saturation, Value*), uma representação frequentemente adotada em tarefas de visão computacional devido à sua separação entre componentes de cor e intensidade luminosa. A Figura 2 ilustra a transformação da imagem original para a escala HSV.



**Figure 2. Transformação de uma imagem da escala RGB para HSV**

Para a aplicação dos filtros e posterior extração das características locais, foi utilizado exclusivamente o canal V (*Value*), correspondente ao brilho da imagem. O uso do canal de brilho permite uma melhor detecção de variações estruturais, independentemente das cores presentes na imagem original.

### 4. Construção dos Filtros

Ao todo, foram empregados cinco filtros distintos: dois filtros para detectar as direções vertical e horizontal, dois filtros para detectar diagonais ascendente e descendente, e um filtro circular.

Cada um desses filtros foi aplicado individualmente em imagens de exemplo para avaliar seu efeito e determinar o impacto de cada um na extração de características de textura. A aplicação individual de cada filtro permite observar como cada tipo de variação é detectado em diferentes direções, seja na vertical, horizontal ou nas diagonais. A Figura 3 demonstra exemplos da aplicação do filtro circular no canal V de algumas imagens.

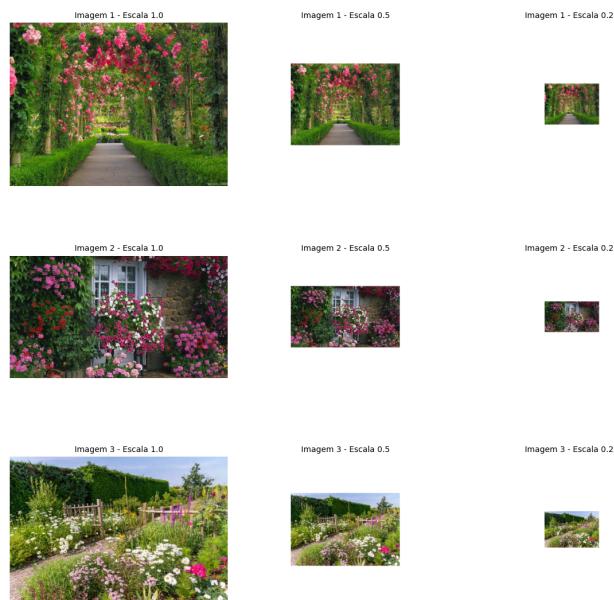
Após a avaliação inicial, os filtros foram aplicados sucessivamente nas imagens, com o intuito de observar como a combinação de diferentes filtros pode realçar características distintas nas imagens. Cada filtro foi aplicado na imagem de forma sucessiva para explorar a possibilidade de extração de múltiplos tipos de características, como bordas e mudanças de intensidade em várias direções.



**Figure 3. Filtro laplaciano aplicado individualmente**

## 5. Escalas

Foram adotadas três escalas dinâmicas para o processamento de cada imagem, com o objetivo de capturar informações texturais em diferentes níveis de detalhe. O procedimento consistiu em aplicar os filtros descritos na Seção anterior sobre a imagem original, correspondente à escala 1.0. Em seguida, foi aplicado um filtro Gaussiano com o intuito de suavizar a imagem, reduzindo ruídos e transições abruptas, o que permite preservar informações relevantes ao reduzir a resolução.



**Figure 4. Diferença entre as escalas de tamanho**

Após a suavização, a imagem foi reduzida pela metade em cada dimensão, originando a escala 0.5. Esse processo foi repetido mais uma vez, gerando uma terceira versão

da imagem na escala 0.25. Assim, para cada imagem de entrada, foram obtidas três versões em diferentes escalas espaciais: 1.0, 0.5 e 0.25, como demonstrado na Figura 4.

## 6. Processamento

### 6.1. Coleta dos Vetores de Características

A etapa de coleta dos vetores de características consiste na extração de informações relevantes que descrevem a textura das imagens em diferentes escalas, utilizando filtros específicos para identificar características espaciais, como bordas e padrões.

Primeiramente, as imagens são reduzidas em tamanho, utilizando diferentes fatores de escala (como 1.0, 0.5 e 0.25), para capturar características em diferentes níveis de resolução. Para cada escala, a imagem passa por uma aplicação de filtro gaussiano para suavizar e remover ruídos, além de reduzir a imagem à metade em cada dimensão, mantendo a proporcionalidade.

Após a redução de escala e o filtro gaussiano, os cinco filtros principais são aplicados às imagens. Para cada imagem filtrada, é extraído um vetor de características, que descreve a textura da imagem em uma janela fixa de 3x3 (tamanho dos filtros construídos). A média da intensidade de cada janela é calculada e armazenada como o valor representativo daquela região. Este vetor contém informações sobre os padrões locais da imagem, como bordas e transições de intensidade, essenciais para a descrição e comparação de texturas. A aplicação dos filtros pode ser vista na Figura 5.



**Figure 5. Filtros aplicados no canal V da imagem**

Para cada imagem e escala, os vetores de características são coletados separadamente. Esses vetores são armazenados em uma estrutura de dados, permitindo o uso posterior para análise e agrupamento das imagens com base nas características extraídas. Esse processo foi supervisionado através do uso de *logs*, como os ilustrados na Figura 6.

```
[LOG] Processando Escala Escala 1.0: Figura 14
[LOG] Figura 14 redimensionada para a escala 1.0 após suavização Gaussiana.
[LOG] Figura 14, Filtro Vertical: Características coletadas. Tamanho do vetor: (96942,)
[LOG] Figura 14, Filtro Horizontal: Características coletadas. Tamanho do vetor: (96942,)
[LOG] Figura 14, Filtro Circular: Características coletadas. Tamanho do vetor: (96942,)
[LOG] Figura 14, Filtro 45 Graus: Características coletadas. Tamanho do vetor: (96942,)
[LOG] Figura 14, Filtro 135 Graus: Características coletadas. Tamanho do vetor: (96942,)
[LOG] Processando Escala Escala 0.5: Figura 14
```

**Figure 6. Exemplo de logs utilizados para controle**

## 6.2. Segmentação

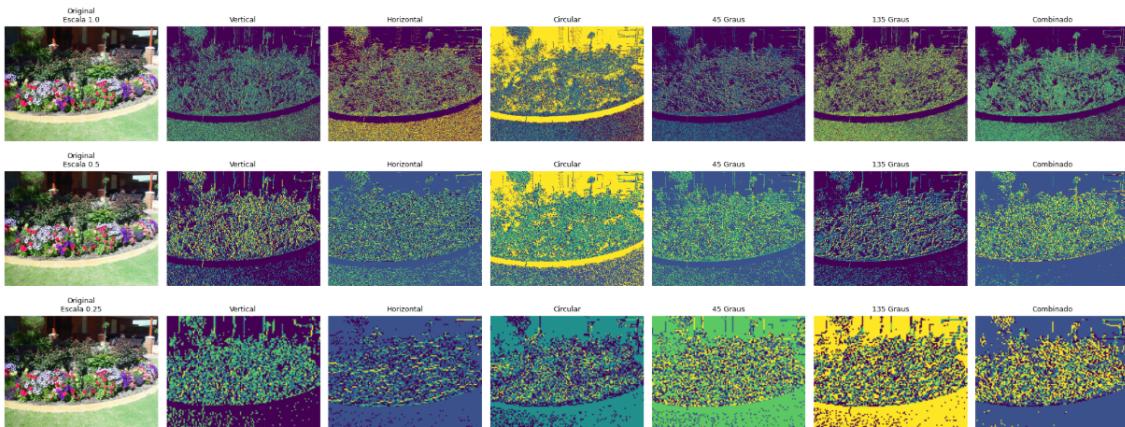
A segmentação por textura foi realizada com base na resposta de filtros convolucionais aplicados ao canal de valor (V) das imagens no espaço de cores HSV. O objetivo é dividir a imagem em regiões com padrões texturais semelhantes, facilitando a análise e a identificação de áreas de interesse.

Inicialmente, as imagens foram suavizadas utilizando um filtro Gaussiano para reduzir ruídos e garantir uma resposta mais estável dos filtros texturais. Em seguida, aplicaram-se cinco filtros convolucionais distintos: horizontal, vertical, diagonal  $45^\circ$ , diagonal  $135^\circ$  e circular (Laplaciano). Cada filtro realça diferentes orientações e padrões de textura presentes na imagem.

A extração de características foi feita por meio da média local das respostas filtradas em janelas de tamanho fixo. Esse processo gerou vetores de textura que foram normalizados e utilizados como entrada para o algoritmo de agrupamento *KMeans*, com um número fixo de clusters  $k = 3$ .

## 7. Resultado Final

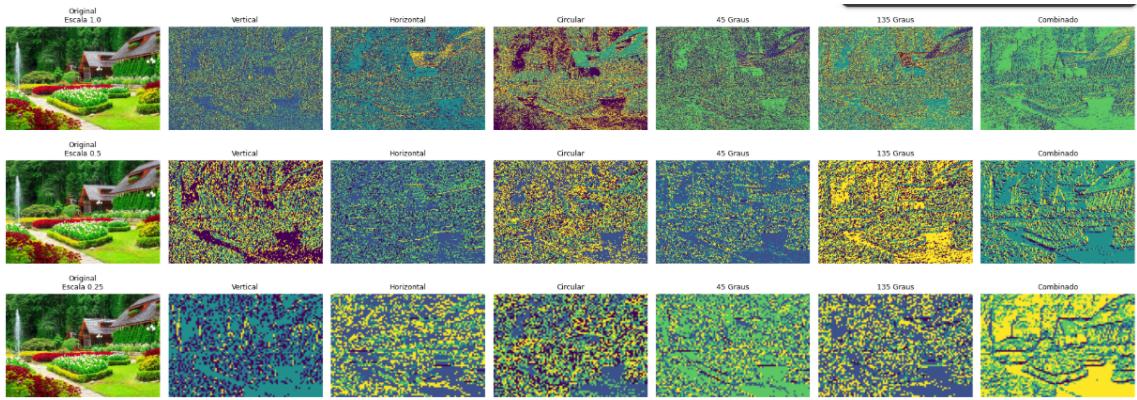
Os resultados foram organizados em visualizações horizontais, em que cada linha representa uma escala aplicada à imagem. Em cada linha, são apresentados, lado a lado, a imagem original redimensionada, as segmentações obtidas por cada filtro individual e a segmentação combinada. Essa disposição facilita a comparação direta entre os filtros e escalas, permitindo uma análise clara do comportamento da segmentação em diferentes condições.



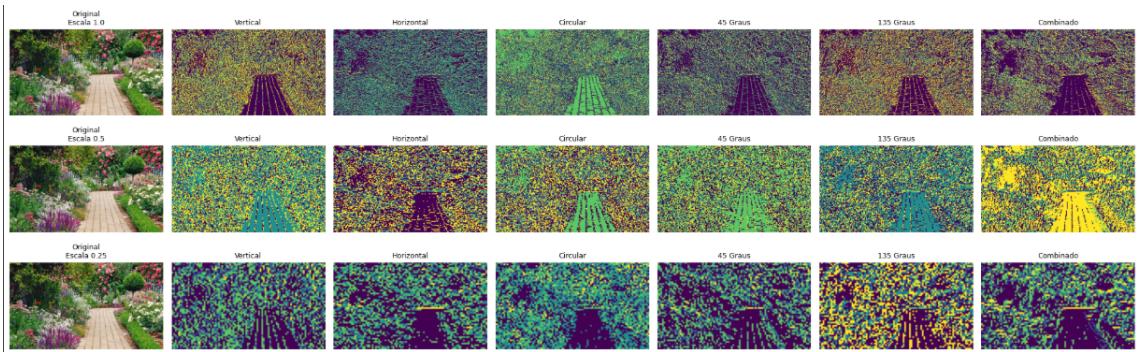
**Figure 7. Exemplo do resultado obtido com a segmentação**

Na escala 1.0, os resultados foram bastante satisfatórios, com boa preservação das estruturas e contornos presentes na imagem original. As regiões segmentadas apresentaram coerência visual, tornando possível distinguir áreas com diferentes padrões texturais.

Entretanto, à medida que a escala diminui (0.5 e 0.25), observa-se uma perda significativa na qualidade da segmentação. Os resultados tornam-se progressivamente mais desconexos e os contornos das regiões segmentadas perdem nitidez, dificultando a interpretação e a associação com as estruturas reais da imagem. As Figuras 7, 8 e 9 demonstram alguns dos resultados obtidos na análise.



**Figure 8. Exemplo do resultado obtido nas diferentes escalas**



**Figure 9. Exemplo do resultado obtido com os diferentes filtros**

## 8. Código Desenvolvido

Todo o código desenvolvido, bem como os resultados obtidos e as imagens analisadas, estão disponíveis no seguinte repositório do *GitHub*: Segmentação De Imagens Por Textura.