

## IMPLEMENTAÇÃO DE UMA VERSÃO DE PERCOLAÇÃO MULTIESCALA E MULTIDIMENSIONAL PARA EXTRAÇÃO DE CARACTERÍSTICAS EM IMAGENS HISTOLÓGICAS

Adão Marcelo Cuevas<sup>1</sup>, Luiz Fernando Segato dos Santos<sup>1</sup>

Instituto Federal do Mato Grosso do Sul – Aquidauana-MS

adao.cuevas@estudante.ifms.edu.br, luiz.santos@ifms.edu.br

### Resumo

Neste trabalho foi realizado um estudo sobre importantes técnicas para extração de características de imagens no contexto da histopatologia. Primeiramente foi realizada uma pesquisa em artigos científicos sobre melhorias nas técnicas básicas e também na busca por novas técnicas. Uma nova técnica encontrada foi a da percolação multiescala e multidimensional. A partir da técnica foi realizado um estudo para entender seu funcionamento e posteriormente implementá-la e validá-la. Algumas imagens fractais foram pesquisadas e coletadas para os testes iniciais. Um pseudocódigo foi encontrado e a partir disso foi trabalhado uma implementação em Python para seguir com os testes.

**Palavras-chave:** Percolação, validação, implementação.

### Introdução

O *machine learning* tem contribuído na obtenção de modelos computacionais que possam auxiliar estes especialistas no reconhecimento de padrões e na classificação de imagens médicas. Diante disso, a busca por técnicas capazes de representar as principais características no auxílio ao diagnóstico é uma prioridade desta área de estudo (LATEEF; SAHRAN; MAKLUMAT, 2019; DING, 2022).

A utilização de técnicas fractais e de entropia na representação de características de imagens histológicas contribuiu em diversas aplicações como por exemplo em (ASSIRATI; 2014).

As técnicas fractais foram exploradas por meio das seguintes métricas: Dimensão Fractal (DF), Lacunaridade (LAC) e percolação (PERC). A DF foi aplicada em diversos contextos como (PANIGRAHI; et al., 2019; DA SILVA; et al., 2021). Este descritor está estritamente ligado à complexidade da textura analisada e obtém a dimensão de estruturas irregulares comuns neste tipo de imagem. A medida representa o quão irregular o objeto de interesse é. A LAC foi aplicada em (ARALICA, 2020) sendo considerada uma medida complementar à DF, uma vez que proporciona uma análise em como o objeto preenche

o espaço em que está contido. Esta medida também está mais ligada à informação espacial, enquanto a DF é uma medida mais estatística.

A PERC por sua vez, é uma medida que se baseia no conceito de percolação de fluidos. A métrica original representa um modelo matemático relacionado à conectividade de elementos em um sistema. Dada uma matriz quadrada e pontos  $a$  e  $b$  nessa matriz, e se houver conectividade entre os pontos de tal forma que todos os pontos adjacentes a estes também estão conectados, esses pontos pertencem a um aglomerado (ESSAM, 1980). Esses pontos simbolizam a passagem do fluido entre os pontos  $a$  e  $b$  e são também designados por poros. Existem diversos modelos de percolação para imagens, mas, em geral, cada aglomerado tem uma probabilidade de se estender de uma extremidade a outra da imagem, se essa probabilidade atingir um determinado limiar então haverá a percolação. Este descritor é importante para as imagens histológicas porque suas texturas apresentam formato poroso, são normalmente compostas por células contidas em um tecido também conhecido por estroma. A percolação foi unificada a abordagem multiescala e multidimensional e aplicada com sucesso em (ROBERTO; et al., 2017).

Com relação às entropias, os modelos mais aplicados foram da entropia de Shannon e da Amostral. A entropia de Shannon foi aplicada principalmente em contextos unidimensionais e no contexto de imagens em tarefas de segmentação, mas recentemente contribuiu para tarefas de classificação de imagens colorretais H&E (DOS SANTOS; et al., 2022). A entropia Amostral também foi aplicada em diversos contextos de imagens médicas, principalmente em tarefas de classificação (DOS SANTOS; et al., 2018). Esta medida, assim como a PERC e a LAC está ligada à estrutura espacial da textura analisada.

A contribuição destas técnicas é de relevância científica para muitas bases de imagens, portanto suas implementações e validações para poderem compor sistemas *Computer Aided Diagnosis* (CAD) seria de grande ajuda aos interessados.

Com base nessas justificativas e mediante o projeto proposto foram realizadas pesquisas por trabalhos

técnicos que apresentassem o código (algoritmo) ou parte deste para fins de aproveitamento.

## Metodologia

Como o primeiro passo, as técnicas descritas anteriormente foram cuidadosamente pesquisadas na Literatura. A partir dos artigos encontrados foi detectada uma nova abordagem de aplicação além da multiescala, a multidimensional (ROBERTO; et al., 2017). Para isso, os esforços de pesquisa foram voltados para essa nova abordagem.

A estratégia foi de pesquisar pelos códigos que também apresentassem a abordagem multidimensional, uma extensão das técnicas fractais e de entropia para imagens coloridas, conforme as considerações de (IVANOVICI; RICHARD; DECEAN, 2009). Durante as pesquisas foi encontrado um código sobre percolação, mais precisamente sobre rotulagem de aglomerados de Hoshen-Kopelman, e a partir desse código que se manteve o foco do projeto a fim de implementá-lo para a linguagem Python e posteriormente executar os devidos testes para buscar a validação do mesmo. Segue o pseudocódigo encontrado:

Pseudocódigo de Hoshen-Kopelman usado para fazer a varredura e a rotulagem das células, por (TOBIN FRICKE, 2004):

### Raster Scan and Labeling on the Grid

```
largest_label = 0;
label = zeros[n_columns, n_rows]
labels = [0:n_columns*n_rows] /* Array containing
integers from 0 to the size of the image. */

for x in 0 to n_columns {
  for y in 0 to n_rows {
    if occupied[x, y] then
      left = label[x-1, y];
      above = label[x, y-1];
      if (left == 0) and (above == 0) then /* Neither
a label above nor to the left. */
        largest_label = largest_label + 1; /* Make a
new, as-yet-unused cluster label. */
        label[x, y] = largest_label;
      else if (left != 0) and (above == 0) then /*
One neighbor, to the left. */
        label[x, y] = find(left);
      else if (left == 0) and (above != 0) then /*
One neighbor, above. */
        label[x, y] = find(above);
```

```
        else /* Neighbors BOTH to the left and
above. */ union(left,above); /* Link the left and
above clusters. */
        label[x, y] = find(left);
      }
    }
  }
```

### Union

```
void union(int x, int y) {
  labels[find(x)] = find(y);
}
```

### Find

```
int find(int x) {
  int y = x;

  while (labels[y] != y)
    y = labels[y];

  while (labels[x] != x) {
    int z = labels[x];
    labels[x] = y;
    x = z;
  }

  return y;
}
```

Após várias tentativas de implementação foi possível chegar uma versão do código em Python e os primeiros testes com imagens realizados não foram muito animadores. Não era possível gerar algumas métricas como por exemplo a matriz de rótulos, porém com muita persistência foi possível corrigir alguns erros dentro do código mostrando uma evolução razoável e podendo assim executar alguns testes.

Segue uma parte da última versão do código em Python que foi implementada:

```
import cv2

import numpy as np

import matplotlib.pyplot as plt

def hoshen_kopelman(image):

  rows, cols = image.shape

  labels = np.zeros((rows, cols), dtype=int)
```

```

next_label = 1

equivalence = {}

# Primeira passagem

for i in range(rows):
    for j in range(cols):
        if image[i, j] != 0:
            left = labels[i, j - 1] if j > 0 else 0
            up = labels[i - 1, j] if i > 0 else 0

            if left == 0 and up == 0:
                labels[i, j] = next_label
                equivalence[next_label] = next_label

                next_label += 1
            elif left != 0 and up == 0:
                labels[i, j] = left
            elif left == 0 and up != 0:
                labels[i, j] = up
            else:
                min_label = min(left, up)

                labels[i, j] = min_label

                if left != up:
                    equivalence[left] = min_label
                    equivalence[up] = min_label

return labels, len(np.unique(labels)) - 1

```

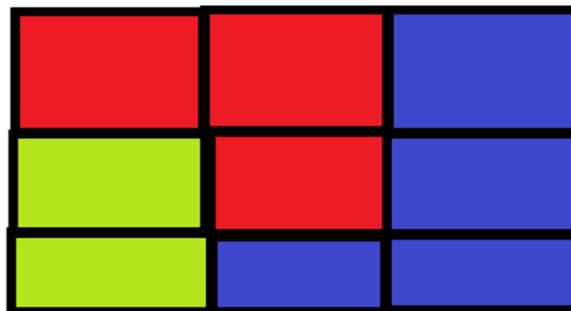


Figura 1 – Imagem criada para teste do código.

```

➡ Número de aglomerados gerados: 10
Matriz de rótulos:
[[1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]]

[[1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]]

Razão Percolante do Aglomerado 1: 0.8865243972273276
Razão Percolante do Aglomerado 2: 0.015289510855634514
Razão Percolante do Aglomerado 3: 0.01605575885820867
Razão Percolante do Aglomerado 4: 0.01580488762604916
Razão Percolante do Aglomerado 5: 0.012194523371927508
Razão Percolante do Aglomerado 6: 0.012816247729888035
Razão Percolante do Aglomerado 7: 0.012933502762310416
Razão Percolante do Aglomerado 8: 0.010422063579496186
Razão Percolante do Aglomerado 9: 0.009484023320117146
Razão Percolante do Aglomerado 10: 0.008475084669040853

```

Figura 2 - Resultado gerado pelo código aplicado na figura 1.

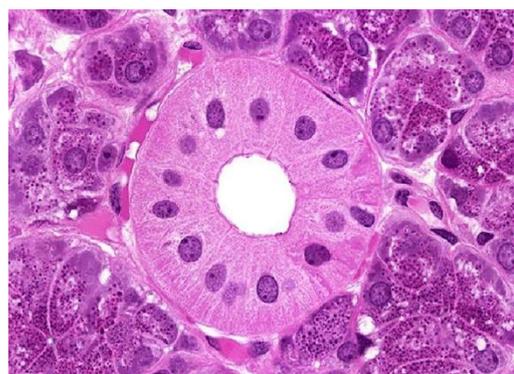


Figura 3 - Ducto estriado

Na tentativa de realizar a validação da técnica seguem algumas imagens e resultados gerados pela aplicação do algoritmo (Figuras 1 a 6).

```
Número de aglomerados gerados: 0
Matriz de rótulos:
[[1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]
 ...
 [1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]]
```

Figura 4 -Resultado gerado pelo código aplicado na figura 3.

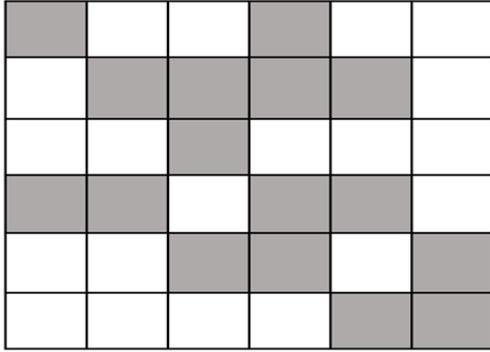


Figura 5 – Imagem criada para teste do código.

```
➡ Número de aglomerados gerados: 28
Matriz de rótulos:
[[1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]
 ...
 [1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]]
Razão Percolante do Aglomerado 1: 0.1572558597309
Razão Percolante do Aglomerado 2: 0.0225953101322
Razão Percolante do Aglomerado 3: 0.0226005176247
Razão Percolante do Aglomerado 4: 0.0229233821622
Razão Percolante do Aglomerado 5: 0.0225901026396
Razão Percolante do Aglomerado 6: 0.0226005176247
Razão Percolante do Aglomerado 7: 0.0226005176247
Razão Percolante do Aglomerado 8: 0.0247876644916
Razão Percolante do Aglomerado 9: 0.0247824569991
Razão Percolante do Aglomerado 10: 0.025141773984
Razão Percolante do Aglomerado 11: 0.024787664491
Razão Percolante do Aglomerado 12: 0.024787664491
Razão Percolante do Aglomerado 13: 0.024787664491
Razão Percolante do Aglomerado 14: 0.0
Razão Percolante do Aglomerado 15: 0.045914461727
Razão Percolante do Aglomerado 16: 0.139050465810
Razão Percolante do Aglomerado 17: 0.045914461727
Razão Percolante do Aglomerado 18: 0.045930084205
Razão Percolante do Aglomerado 19: 0.0
Razão Percolante do Aglomerado 20: 0.0
```

Figura 6 - Resultado gerado pelo código aplicado a figura 5.

Para a aplicação da técnica nas imagens, foram escolhidas métricas como as especificadas em

(ROBERTO; et al., 2017), a saber o número médio de aglomerados ( $C(L)$ ) por caixa de tamanho  $L$ , a quantidade de caixas percolantes ( $Q(L)$ ) para cada escala  $L$  e a cobertura do maior aglomerado por caixa ( $M(L)$ ) para cada caixa de lado  $L$ .

De acordo com esses mesmos autores, uma etapa para o cálculo da percolação é a rotulagem de aglomerados, normalmente implementada via algoritmo de Hoshen-Kopelman. Foi feito um estudo minucioso com o código verificando se a rotulagem e cada métrica eram contempladas, e a partir disso foram realizadas diversas tentativas de adaptação do código para calcular a métrica  $C(L)$  que é responsável por calcular o número de caixas por aglomerado.

### Resultados Esperados

Os resultados esperados para este trabalho de pesquisa, implementação e validação são:

1. Conseguir fazer que o código imprima os resultados esperados a fim de poder validar a técnica e posteriormente dar sequência no trabalho;
2. Um repositório (Github) de técnicas multiescalas implementadas e validadas para quantificação e extração de características de imagens histológicas da área médica, somando as novas técnicas implementadas as implementadas anteriormente;
3. O entendimento detalhado da relação entre as propriedades histológicas de imagens representativas de tecidos saudáveis e cancerígenos, e as técnicas implementadas;
4. O desenvolvimento de novos métodos, a partir dos TCCs e iniciação científica, com resultados significativos que utilizem as técnicas implementadas.

### Considerações Finais

O trabalho proposto considerou um amplo levantamento de técnicas fractais, seu entendimento para implementação e validação e a busca de códigos prontos ou semi-prontos para análises e adequações. Novos testes serão realizados posteriormente para os devidos ajustes necessários para o perfeito

funcionamento do código, gerando assim os resultados esperados.

### Agradecimentos

Ao IFMS pela oportunidade de estar enriquecendo essa jornada acadêmica.

### Referências

ARALICA, Gorana et al. Prognostic Significance of Lacunarity in Preoperative Biopsy of Colorectal Cancer. **Pathology & Oncology Research**, v. 26, n. 4, p. 2567-2576, 2020.

ASSIRATI, L. **Entropia aplicada ao reconhecimento de padrões em imagens**. 2014. 120 p. Dissertação (Mestrado em Ciências) Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2014.

DA SILVA, Lucas Glaucio et al. Fractal dimension analysis as an easy computational approach to improve breast cancer histopathological diagnosis. **Applied Microscopy**, v. 51, n. 1, p. 1-9, 2021.

DING, Saisai et al. Fractal Graph Convolutional Network with MLP-mixer based Multi-path Feature Fusion for Classification of Histopathological Images. **Expert Systems with Applications**, p. 118793, 2022.

DOS SANTOS, Luiz Fernando Segato et al. Multidimensional and fuzzy sample entropy (SampEnMF) for quantifying H&E histological images of colorectal cancer. **Computers in biology and medicine**, v. 103, p. 148-160, 2018.

DOS SANTOS, Luiz Fernando Segato et al. Multidimensional shannon entropy (H M) as an approach to classify H&E colorectal images. In: **2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP)**. IEEE, 2022. p. 1-4.

ESSAM, John W. Percolation theory. **Reports on progress in physics**, v. 43, n. 7, p. 833, 1980.

IVANOVICI, M.; RICHARD, N.; DECEAN, H. Fractal dimension and lacunarity of psoriatic lesions- a colour approach. **medicine**, v. 6, n. 4, p. 7, 2009.

FRICKE, Tobin. **Hoshen–Kopelman algorithm**. <https://en.wikipedia.org/https://en.wikipedia.org/>. 2023. Disponível em: [https://en.wikipedia.org/wiki/Hoshen%E2%80%9393Kopelman\\_algorithm](https://en.wikipedia.org/wiki/Hoshen%E2%80%9393Kopelman_algorithm). Acesso em: 11 out. 2023.

LATEEF, A.; SAHRAN, Shahnorbanun; MAKLUMAT, Sains. **Histopathology image classification based on color space**. Tech. rep., Universiti Kebangsaan Malaysia. <https://www.ftsm>.

ukm.  
[my/cybersecurity/file/research/technicalreport/PS-FTSM-2019-002.pdf](https://my.cybersecurity/file/research/technicalreport/PS-FTSM-2019-002.pdf), 2019.

PANIGRAHI, Santisudha et al. Fractal Geometry for Early Detection and Histopathological Analysis of Oral Cancer. In: **International Conference on Mining Intelligence and Knowledge Exploration**. Springer, Cham, 2019. p. 177-185.

## IMPLEMENTATION OF A MULTISCALE AND MULTIDIMENSIONAL PERCOLATION VERSION FOR FEATURE EXTRACTION IN HISTOLOGICAL IMAGES

**Abstract:** *In this work, a study was conducted on important techniques for feature extract from images in the context of histopathology. First a research in scientific papers about improvements in the basic techniques and also in the search for new techniques was performed. A new technique found was the multiscale and multidimensional percolation. Based on this technique a study was done to understand how it works and later implement and validate it. Some fractal images were researched and collected for initial tests. An implementation of the technique in Python was also found on the internet and some initial tests on the images found were done for an initial validation.*

**Keywords:** .Percolation, validation, implementation.