



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA

TI0147 - FUNDAMENTOS DE PROCESSAMENTO DIGITAL DE IMAGENS

**PROCESSAMENTO DIGITAL DE IMAGENS PARA A SEGMENTAÇÃO DE
PULMÕES EM IMAGENS DE TOMOGRAFIA COMPUTADORIZADA**

FORTALEZA
2025

Resumo—Neste artigo, foi desenvolvida uma abordagem para segmentação automática de pulmões em imagens de Tomografia Computadorizada (TC), utilizando o Algoritmo de Contornos Ativos Crisp Adaptativo 2D. O objetivo foi otimizar a identificação das estruturas pulmonares por meio de cálculos de energia interna e externa, reduzindo a necessidade de segmentação manual. A metodologia empregada envolveu a análise das densidades dos pixels em Unidades Hounsfield (UH), a definição de regiões iniciais e a evolução iterativa da curva de segmentação até a convergência do contorno.

Os resultados demonstraram a eficiência do algoritmo na delimitação precisa dos pulmões, garantindo segmentações estáveis e ajustadas às características anatômicas. Apesar das limitações inerentes ao método, como a necessidade de ajustes em casos específicos de variação de contraste, a proposta contribui significativamente para o processamento automatizado de imagens médicas. Sua aplicação pode auxiliar em diagnósticos e estudos clínicos, tornando o fluxo de análise mais eficiente e menos suscetível a erros humanos.

Palavras-chave—Pulmão, segmentação, imagem, algoritmo, TC.

I. INTRODUÇÃO

A. Imagens Digitais

Uma imagem digital é formada por pequenos pontos chamados pixels. Cada pixel tem um valor que representa sua cor e a sua análise varia a depender da escala de cores que está sendo utilizada. Em imagens em preto e branco, esse valor indica o tom de cinza, indo do preto ao branco. Já em imagens coloridas, cada pixel combina três cores básicas (vermelho, verde e azul) para formar diferentes tonalidades. Dessa forma, ao atribuir-se um valor numérico a cada pixel de uma imagem, é possível convertê-la numa Matriz, permitindo armazenar e manipular matematicamente os valores dela.

B. Transformações lineares

Tendo ciência de que cada pixel de uma imagem possui um valor agregado que descreve as características daquele ponto, é possível entender que, numericamente, o ajuste de brilho de uma imagem pode ser realizado através de uma operação de adição em cada pixel da imagem, porquanto o aumento dos valores na escala desloca o tom para valores mais altos. Matematicamente, essa transformação pode ser expressa como:

$$g(x, y) = f(x, y) + b, \quad (1)$$

no qual:

- $f(x, y)$ é o valor do pixel original na posição (x, y) ;
- b é o valor da alteração de brilho (positivo para aumentar o brilho, negativo para reduzir);
- $g(x, y)$ é o valor do pixel modificado.

O ajuste de contraste pode ser feito através de uma multiplicação linear dos valores de intensidade dos pixels por um fator α , pois a multiplicação melhora o contraste já que amplia a diferença entre os tons claros e escuros, fazendo com que áreas escuras fiquem ainda mais escuras e áreas claras mais brilhantes expressa por:

$$g(x, y) = \alpha f(x, y). \quad (2)$$

Neste caso:

- $\alpha > 1$ aumenta o contraste;
- $0 < \alpha < 1$ reduz o contraste.

Ajustes de brilho e contraste podem ser combinados:

$$g(x, y) = \alpha f(x, y) + b. \quad (3)$$

C. Limiarização

A limiarização, ou thresholding, é uma técnica essencial no processamento digital de imagens usada para segmentação, removendo objetos indesejados e separando o fundo da imagem. O processo consiste em transformar uma imagem em escala de cinza em uma imagem binária, fazendo com que os pixels assumam apenas dois valores: preto (0) ou branco (1). De tal forma que:

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) \geq T \\ 0, & \text{se } f(x, y) < T \end{cases}, \quad (4)$$

Sendo:

- $f(x, y)$ é o valor de intensidade do pixel na posição (x, y) ;
- T é o valor do limiar (threshold);
- $g(x, y)$ é o valor do pixel resultante na imagem binária.

Se o valor do pixel original for maior ou igual ao limiar, ele será transformado em branco (1), caso contrário, será transformado em preto (0). Essa abordagem é útil para destacar regiões de interesse em uma imagem e remover detalhes irrelevantes. Parte do processo envolve uma afinidade com a técnica para utilização de um parâmetro que seja adequado.

Existem variações mais eficientes desse método, como a limiarização adaptativa, de tal forma que o limiar T pode variar de acordo com regiões da imagem, e a limiarização com histograma, que define T com base na distribuição dos tons de cinza da imagem. Dessa forma, é possível manipular especificamente as regiões de interesse na imagem.

D. Segmentação Baseada em Contornos Ativos

A segmentação por contornos ativos é uma abordagem iterativa que refina o contorno de um objeto na imagem. Esse método se baseia na minimização de uma função de energia que equilibra a suavidade do contorno e a aderência às bordas da imagem.

A equação geral da energia total do contorno ativo é dada por:

$$E = \int (E_{interno} + E_{externo}) ds, \quad (5)$$

De tal forma que:

- $E_{interno}$ representa a energia interna do contorno, que controla sua suavidade;
- $E_{externo}$ representa a energia externa, que atrai o contorno para os limites da região de interesse;
- ds é o elemento diferencial ao longo do contorno.

A energia interna é composta por termos que regulam a continuidade e a curvatura do contorno:

$$E_{interno} = \alpha \left| \frac{dC(s)}{ds} \right|^2 + \beta \left| \frac{d^2C(s)}{ds^2} \right|^2, \quad (6)$$

nela, α controla a elasticidade e β controla a rigidez do contorno.

A energia externa é baseada nas características da imagem, como gradientes de intensidade:

$$E_{\text{externo}} = -|\nabla I(x, y)|^2, \quad (7)$$

$I(x, y)$ representam a intensidade da imagem e $\nabla I(x, y)$ é seu gradiente.

O contorno é ajustado iterativamente minimizando E , deslocando seus pontos na direção do gradiente descendente:

$$\frac{\partial C(s, t)}{\partial t} = -\nabla E. \quad (8)$$

Dessa forma, permite-se que o contorno inicial evolua até se alinhar às bordas da estrutura desejada.

E. Algoritmo de Contornos Ativos Crisp Adaptativo 2D

O Algoritmo de Contornos Ativos Crisp é uma variação dos modelos de contornos ativos usados para segmentação de imagens médicas. Ele busca encontrar as bordas de um objeto minimizando uma função de energia composta por termos internos e externos. O termo de energia interna controla a suavidade do contorno, enquanto a energia externa atrai o contorno para as bordas da imagem, utilizando gradientes de intensidade.

Matematicamente, o contorno é representado por uma curva e sua evolução ocorre minimizando a função de energia, dada por:

$$E_{\text{total}} = \int_0^L \left(\alpha |C'(s)|^2 + \beta |C''(s)|^2 - \gamma |\nabla I(x, y)|^2 \right) ds. \quad (9)$$

Os parâmetros da função de energia são definidos de acordo com a aplicação específica e as características da imagem. O coeficiente α controla a tensão do contorno, ou seja, sua resistência a mudanças bruscas de forma, evitando oscilações excessivas. Já o coeficiente β regula a rigidez do contorno, garantindo que ele permaneça suave e não forme dobras abruptas. O parâmetro γ ajusta a influência do gradiente da imagem, garantindo que o contorno seja atraído para regiões de borda mais pronunciadas. A evolução da curva ocorre iterativamente segundo a equação diferencial:

$$\frac{\partial C}{\partial t} = -\nabla E_{\text{total}}. \quad (10)$$

Resolver essa equação significa calcular a direção de maior redução da energia total e mover o contorno nessa direção. Como a energia externa é negativa nas regiões de borda (pois deriva do gradiente da imagem), a minimização da energia total leva a curva a se alinhar automaticamente com as descontinuidades de intensidade na imagem. Além disso, a energia interna garante que o contorno permaneça suave e contínuo, evitando oscilações que poderiam levar a segmentações imprecisas.

F. Pulmão

Os pulmões são órgãos vitais responsáveis pela troca gasosa no organismo, permitindo a absorção de oxigênio e a eliminação do dióxido de carbono. Eles possuem uma estrutura altamente complexa, composta por brônquios, bronquíolos e alvéolos, que garantem a eficiência desse processo. Devido à sua exposição constante a agentes externos, como poluentes, vírus e bactérias, os pulmões estão suscetíveis a uma série de doenças que comprometem sua função e impactam a saúde respiratória.

No contexto da análise de imagens médicas, algumas das principais doenças pulmonares detectáveis por Tomografia Computadorizada (TC) incluem: Doença Pulmonar Obstrutiva Crônica (DPOC), Câncer de Pulmão, Pneumonia, Tuberculose, Fibrose Pulmonar, COVID-19 e Outras Infecções Virais.

A segmentação automática de pulmões em imagens médicas facilita a detecção e o acompanhamento dessas doenças, permitindo uma análise mais precisa e objetiva das estruturas pulmonares. Durante a análise, as imagens obtidas possuem um quantitativo numérico, a escala de Hounsfield, que avalia a densidade das estruturas do corpo humano, permitindo o uso de recursos como o auxílio da inteligência artificial e do processamento digital de imagens, permitindo acelerar diagnósticos, monitorar a progressão de patologias e otimizar a tomada de decisões clínicas, resultando em melhores prognósticos e tratamentos mais eficazes para os pacientes.

II. BACK-END

1) *Algoritmo 1*: A implementação da primeira etapa do algoritmo apresentou resultados satisfatórios na definição das regiões de interesse e na classificação das áreas pulmonares com base nas unidades Hounsfield (UH). A seguir, discutimos os principais achados de cada fase.

A segmentação inicial das regiões esquerda e direita da imagem foi bem-sucedida, com as delimitações nos intervalos de pixels esperados: $[0, 256]$ para a região esquerda e $[256, 512]$ para a região direita no eixo das abcissas, e $[180, 360]$ no eixo das ordenadas. Essa separação permitiu uma análise local mais precisa para cada região pulmonar, garantindo que os objetos de interesse estivessem corretamente centralizados.

A classificação das áreas da imagem em seis classes de pixels foi realizada com sucesso. A matriz de densidades pulmonares $D(x, y)$ possibilitou a quantificação de cada classe com base na distribuição de UH. Os resultados indicaram que a função de contagem $f(x, y, u_i)$ proporcionou uma classificação coerente, permitindo calcular a proporção de cada tipo de tecido pulmonar.

A aplicação do operador de Sobel na determinação da energia externa $E_{\text{crisp}}(x, y)$ mostrou-se eficiente na detecção de bordas relevantes na imagem. Os limiares empíricos adotados ($P_3 < 0,2$ e $P_4 \leq 0,15$) foram adequados para a caracterização das regiões de interesse, resultando em uma segmentação coerente e condizente com os padrões esperados.

Os pontos de inicialização das curvas foram corretamente identificados com base na função $C_{\text{crisp}}(x, y)$. A determinação das coordenadas iniciais para as curvas esquerda

(x_e, y_e) e direita (x_d, y_d) permitiu a formação de polígonos regulares centrados nessas posições. A distribuição dos 30 pontos ao longo da curva inicial manteve a consistência geométrica, assegurando uma boa correspondência com os contornos pulmonares reais.

Os resultados obtidos nesta etapa indicam que a definição das regiões e a classificação das áreas pulmonares foram implementadas de forma robusta, tendo em vista tratar-se das definições iniciais para a execução do próximo algoritmo, que efetuará a delimitação da região pulmonar. A identificação dos pontos de inicialização e a geração das curvas iniciais foram consistentes com as expectativas. Essas bases são fundamentais para o sucesso das próximas etapas do algoritmo de contornos ativos.

2) *Algoritmo 2*: Uma vez inicializada a curva, será iniciado um ciclo iterativo de passos para remover, adicionar e movimentar os pontos da curva, até que ela tome a forma do pulmão (idealmente). Os passos iterativos são descritos nas seções seguintes.

Os pontos da curva serão movimentados de forma a minimizar uma função de energia E da curva, ponto a ponto. Isto é possível analisando a vizinhança de cada ponto, calculando as energias envolvidas e movendo-o para a coordenada que possuir a menor energia total E na vizinhança do ponto. Aqui, será analisada a vizinhança 9×9 do ponto.

A energia total de um ponto $c(s)$ é dada por:

$$E_{\text{total}}[c(s)] = E_{\text{intadap}}[c(s)] + E_{\text{crisp}}, \quad (11)$$

em que E_{intadap} é a energia interna e E_{crisp} é a energia externa. A energia externa de cada ponto foi calculada durante a inicialização, e a energia interna será detalhada a seguir.

A energia interna é definida como:

$$E_{\text{intadap}}[c(s)] = w_{\text{cont}} F_{\text{cont}}[c(s)] + w_{\text{adap}} F_{\text{adap}}[c(s)], \quad (12)$$

em que $F_{\text{cont}}[c(s)]$ é a Força de Continuidade, $F_{\text{adap}}[c(s)]$ é a Força Balão Adaptativa e os parâmetros w_{cont} e w_{adap} são pesos associados a cada força para ajustar a importância de cada um dos seus respectivos termos no cálculo da energia interna E_{intadap} . Neste caso, os valores são $w_{\text{cont}} = 0.6$ e $w_{\text{adap}} = 0.1$.

A Força de Continuidade $F_{\text{cont}}[c(s)]$ é dada por:

$$F_{\text{cont}} = DM - \sqrt{(x(s) - x(s-1))^2 + (y(s) - y(s-1))^2}, \quad (13)$$

sendo o termo à direita a distância euclidiana entre dois pontos consecutivos da curva (ponto s e ponto $s-1$), e DM é dado por:

$$DM = \frac{\sum_{i=0}^{N-1} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{N}, \quad (14)$$

em que N é a quantidade de pontos presentes atualmente na curva e i é o índice de ordenação do ponto na curva.

Já a Força Balão Adaptativa $F_{\text{adap}}[c(s)]$ é definida como:

$$F_{\text{adap}}[c(s)] = \sqrt{E_x^2 + E_y^2}, \quad (15)$$

em que as componentes E_x e E_y são definidas por:

$$E_x = |x(s) \pm x_m|; \quad (16)$$

$$E_y = |y(s) \pm y_m|, \quad (17)$$

em que $x(s)$ e $y(s)$ são as coordenadas do ponto $c(s)$ na qual F_{adap} está sendo calculada, e x_m e y_m são as coordenadas do ponto médio dos pontos vizinhos imediatos. O sinal utilizado nas equações das componentes E_x e E_y é determinado pela topologia da curva, sendo negativo quando o ponto médio dos vizinhos imediatos é externo à curva e positivo quando o ponto está interno.

O método de remoção de pontos se baseia em dois critérios. O primeiro consiste em apagar o ponto quando suas coordenadas são iguais às coordenadas de um ponto consecutivo. Já o segundo critério se baseia na remoção de pontos que possuem variação brusca de coordenadas em relação aos seus vizinhos. Esta variação é detectada a partir do ângulo formado entre o ponto analisado e seus vizinhos. Este ângulo α é determinado utilizando a lei dos cossenos dado por:

$$\alpha = \arccos \left(\frac{a^2 + b^2 - c^2}{2ab} \right). \quad (18)$$

O método da adição de pontos se baseia na distância Euclidiana entre dois pontos consecutivos do contorno, em que um ponto é adicionado quando a distância entre dois pontos consecutivos é maior do que uma distância máxima configurada previamente.

Considerando dois pontos vizinhos com uma distância maior que a máxima configurada sejam os pontos V_1 e V_2 , compostos pelas coordenadas (x_{V_1}, y_{V_1}) e (x_{V_2}, y_{V_2}) , respectivamente. O ponto médio $M(x_M, y_M)$ desta semi-reta é dado por:

$$y_M = \frac{y_{V_1} + y_{V_2}}{2}; \quad (19)$$

$$x_M = \frac{x_{V_1} + x_{V_2}}{2}. \quad (20)$$

O ângulo α desta semi-reta em relação à linha horizontal é definido como:

$$\alpha = \arctan \left(\frac{y_{V_1} - y_{V_2}}{x_{V_1} - x_{V_2}} \right). \quad (21)$$

Para determinar a direção citada, define-se dois pontos $P_1(x_1, y_1)$ e $P_2(x_2, y_2)$ como:

$$y_1 = y_M + d \times \sin(\alpha + \frac{\pi}{2}); \quad (22)$$

$$x_1 = x_M + d \times \cos(\alpha + \frac{\pi}{2}); \quad (23)$$

$$y_2 = y_M + d \times \sin(\alpha - \frac{\pi}{2}); \quad (24)$$

$$x_2 = x_M + d \times \cos(\alpha - \frac{\pi}{2}). \quad (25)$$

O ponto a ser adicionado é:

$$y_A = y_M + d \times \sin(\alpha_a); \quad (26)$$

$$x_A = x_M + d \times \cos(\alpha_a). \quad (27)$$

A curva é rotulada como estável quando o perímetro não aumenta após duas iterações consecutivas.

3) *Implementação e Otimização*: A implementação do algoritmo foi desenvolvida em Python, utilizando a biblioteca NumPy para operações numéricas eficientes. O NumPy foi escolhido por sua capacidade de realizar operações vetorizadas, substituindo loops nativos do Python por operações otimizadas em C, o que resulta em um ganho significativo de desempenho. Além disso, para otimizar ainda mais o processamento, foi utilizada a biblioteca Numba, que compila trechos críticos do código em tempo de execução (JIT - Just-In-Time) para código de máquina, proporcionando uma execução mais rápida, especialmente em operações repetitivas e de grande escala.

Essa combinação de vetorização com NumPy e compilação JIT com Numba permitiu alcançar um desempenho próximo ao de linguagens de baixo nível, mantendo a simplicidade e a produtividade do Python.

4) *Tempo de Processamento*: Foram realizadas análises de desempenho do algoritmo, considerando tanto a versão sem otimizações quanto uma versão otimizada utilizando a biblioteca Numba.

Na versão sem otimizações específicas, os tempos de execução observados foram os seguintes para 50 iterações da minimização da curva:

- **Tempo total da minimização**: 53,8 segundos
- **Tempo gasto na função que calcula a Força de Continuidade**: 31,7 segundos
- **Tempo gasto na função que calcula a Força Adaptativa**: 17,2 segundos

As funções *adicionar pontos* e *remover pontos* apresentaram tempos de execução irrelevantes em comparação à minimização da curva, sendo ordens de magnitude mais rápidas. Cabe destacar que a minimização refere-se exclusivamente ao cálculo da nova curva, sem adição ou remoção de pontos.

Após a aplicação de otimizações utilizando o Numba, obteve-se o tempo total da minimização para 50 iterações de 73,85 segundos, do qual o tempo correspondente à compilação do Numba é 73,6 segundos. É importante ressaltar que a compilação realizada pelo Numba ocorre apenas na primeira execução do programa. Dessa forma, a primeira iteração do algoritmo apresenta um tempo de execução maior, enquanto as iterações subsequentes são significativamente mais rápidas.

Ao considerar apenas a execução do código já compilado, foi obtido um speedup de 21.550%. Além disso, verificou-se que, à medida que o número de iterações aumenta, o tempo de execução do algoritmo também cresce, devido ao acréscimo de novos pontos ao longo do processo.

Outro aspecto relevante é que a versão otimizada com Numba faz uso de paralelismo para a minimização da curva.

Consequentemente, sem otimizações, o tempo de convergência tende a aumentar ainda mais, enquanto, com a utilização do Numba, o tempo de execução permanece mais estável.

5) *Execução do Algoritmo - Exemplo*: Foram conduzidos diversos testes utilizando as imagens de pulmões disponibilizadas pelo professor. A figura 1 ilustra o resultado da segmentação aplicada a uma das imagens.

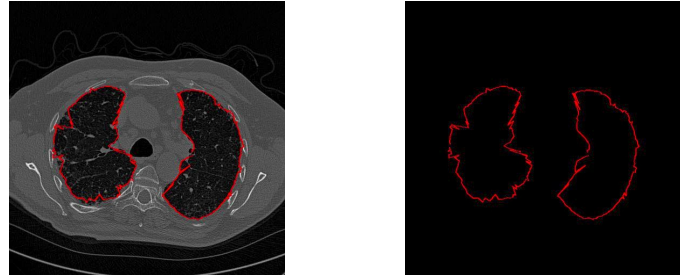


Figura 1: Segmentação utilizando o Algoritmo de Contornos Ativos Crisp Adaptativo 2D

Ao analisar a figura 1, observa-se que o algoritmo foi capaz de delimitar corretamente a maior parte da estrutura pulmonar, demonstrando a eficácia da implementação na segmentação da imagem. Esse resultado indica que o método empregado atende aos critérios estabelecidos, possibilitando a identificação adequada das regiões de interesse. Contudo, é relevante considerar possíveis limitações ou ajustes necessários para aprimorar a precisão da segmentação em casos mais complexos.

6) *Métodos Alternativos*: Embora o Algoritmo de Contornos Ativos Crisp Adaptativo 2D tenha se mostrado eficiente para a segmentação de pulmões em imagens de Tomografia Computadorizada (TC), a implementação de métodos alternativos é fundamental para aprimorar a precisão e a robustez do processo. Além disso, algumas outras técnicas de segmentação podem oferecer vantagens em cenários específicos, lidando melhor com variações de contraste, ruído e heterogeneidade dos tecidos pulmonares, por exemplo. Foram implementados e testados então alguns métodos complementares para comparação de desempenho. Cada uma das técnicas a seguir apresenta particularidades que podem influenciar na qualidade da segmentação final e na adaptabilidade do processo a diferentes condições de imagem. Para comparação com os demais métodos, o resultado obtido com a implementação do algoritmo está a seguir.

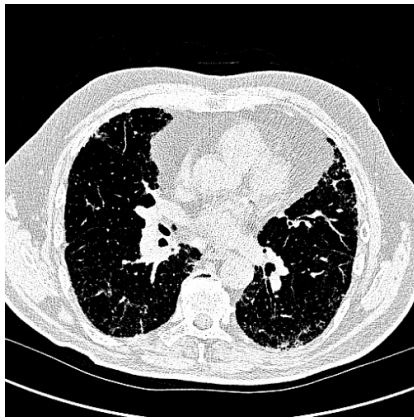


Figura 2: Método Principal

- Crescimento de regiões com a semente fora do pulmão: Agrupa regiões que compartilham propriedades parecidas, como intensidade de pixels, textura ou cor. O método é iniciado a partir de um ou mais pixels denominados de “sementes”, que podem pertencer ou não ao elemento que se deseja segmentar. São definidos critérios de similaridade, que indicam quais pixels devem ser incluídos na região baseados em sua textura, intensidade ou cor. A partir das sementes e dos critérios de similaridade é iniciado o crescimento da região, no qual os pixels que atendem aos critérios são adicionados a região, parando quando não há mais pixels disponíveis a serem adicionados ou o tamanho máximo da região foi atingido.

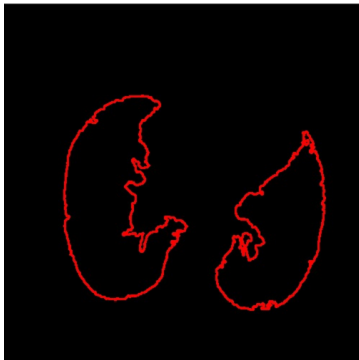


Figura 3: Crescimento de regiões com a semente fora do pulmão

- Divisão e fusão de regiões: Consiste em fazer divisões na imagem principal e agrupar os blocos formados dessas divisões baseado em um critério de junção, caso o critério falhe, as regiões são novamente divididas até que se chegue a uma região de tamanho unitário. Critérios de homogeneidade escolhidos: 1 - Variância de intensidades: checa se a variância da região considerada excede o limite; 2 - Média de intensidades: checa se a diferença entre a média da região e um valor referência excede um limite; Caso ambos os critérios sejam satisfeitos, a região não é dividida; se ao menos um não satisfizer, ela

é novamente dividida e os critérios são avaliados para cada nova região criada.

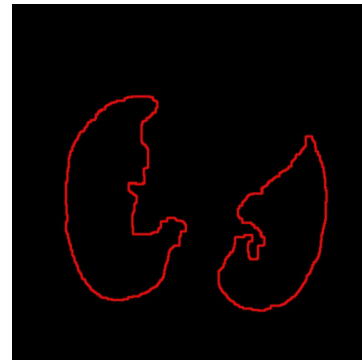


Figura 4: Divisão e fusão de regiões

- Watershed: Aplica o algoritmo watershed para segmentação. Utiliza o conceito de bacias hidrográficas (watersheds), usando uma topologia de três dimensões: duas coordenadas espaciais x intensidade. Consideram-se três tipos de pontos: os pontos que pertencem a um mínimo regional; pontos em que uma gota d'água, se despejada na localização de algum desses pontos, cairia quase que com certeza em um mínimo simples; e pontos em que a água tem a mesma probabilidade de cair em mais de um mínimo desses (linhas de divisão). O objetivo do algoritmo é encontrar essas linhas.



Figura 5: Watershed

- Limiarização com média móvel: A limiarização com média móvel é uma técnica de segmentação adaptativa que ajusta dinamicamente o limiar para cada pixel com base na média dos valores de intensidade em uma vizinhança local. Esse método é útil para imagens com variações suaves de iluminação. O processo é realizado da seguinte forma: para cada pixel, calcula-se a média dos pixels vizinhos dentro de uma janela $N \times N$, após isso, o limiar $T(x,y)$ é definido como essa média local e o pixel é classificado como fundo ou objeto comparando sua intensidade com $T(x,y)$. Funciona bem para imagens com iluminação não uniforme e é um método simples e eficiente computacionalmente. É comumente utilizada

quando Otsu ou limiares fixos falham devido a variações de iluminação.

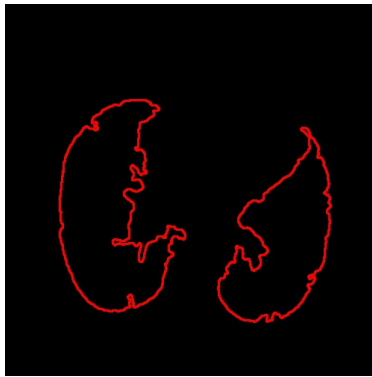


Figura 6: Limiarização com média móvel

- Limiarização múltipla: Divide a imagem em mais de duas classes usando múltiplos limiares. Diferente da limiarização simples (um único T), ela define vários valores (T_1, T_2, \dots, T_n) para separar diferentes regiões da imagem. Ela apresenta uma melhor segmentação para imagens com múltiplas regiões de interesse e funciona bem quando o histograma apresenta múltiplos picos, no entanto, é sensível a ruídos e pode exigir mais tempo de processamento.

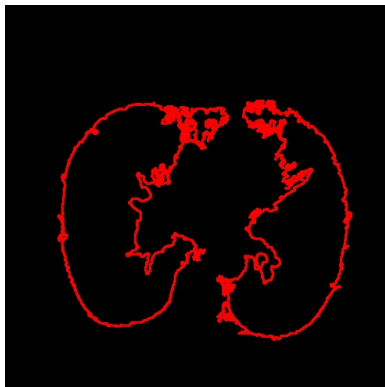


Figura 7: Limiarização Múltipla

- Limiarização com propriedades locais: Técnica de segmentação adaptativa que define o limiar de cada pixel com base em características da sua vizinhança, ao invés de usar um único limiar global. Esse método é útil para imagens com variações não uniformes de iluminação. Funciona bem em imagens com iluminação não uniforme e segmenta melhor regiões com variações suaves de intensidade.

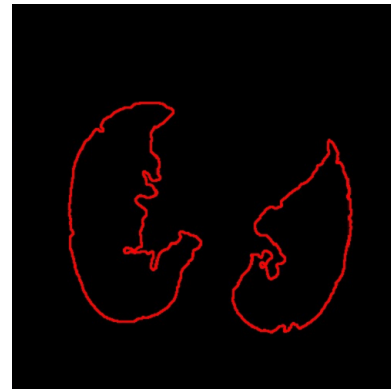


Figura 8: Limiarização com propriedades locais

- Otsu: O método é uma variação da técnica de limiarização em imagens tons de cinza. Ele calcula um limiar ótimo minimizando a variância intra-classe (dentro das regiões) e maximizando a variância inter-classe (entre as regiões). É particularmente eficaz para imagens bimodais, onde há dois picos bem definidos no histograma.



Figura 9: Otsu

- Savoula: É uma técnica de limiarização local útil para imagens que não são uniformes. Ao invés de calcular um único limiar global para a imagem inteira, vários limiares são calculados para todos os pixels utilizando fórmulas específicas que levam em conta a média e o desvio-padrão para cada vizinhança local.

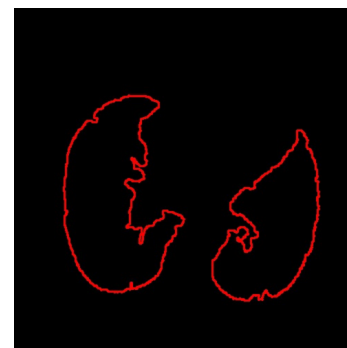


Figura 10: Savoula

III. FRONT-END

1) *Protótipo de UI*: O protótipo da interface foi desenvolvido no Figma, seguindo o escopo do projeto. Ele inclui:

- Seleção e Visualização: Escolha de imagens DICOM e exibição no painel principal;
- Segmentação: Ferramentas para contorno manual e visualização da segmentação automática;
- Controles: Ajuste de zoom e parâmetros do algoritmo;
- Exportação: Download da imagem segmentada e coordenadas dos pontos;
- Acessibilidade: Tema de alto contraste e suporte a português/inglês.

O protótipo serviu como base para a implementação no React, garantindo consistência visual e compatibilidade com a API Flask, que executará os cálculos da segmentação.

2) *Interface*: Com a utilização da biblioteca React, a interface foi implementada com base no protótipo da UI. Recursos não funcionais como modo claro-escuro além de selecionar o idioma entre português-inglês foram implementados, a fim de trazer uma melhor experiência para o usuário. A componentização, aspecto importante para poder reutilizar os componentes em vários locais foi um dos pontos importantes durante todo o processo de implementação. Ao selecionar o arquivo, a interface não permite a escolha de um arquivo de são seja do tipo DICOM, já que é o arquivo esperado no back-end. Na interface é possível a seleção manual, onde podemos segmentar manualmente o pulmão, além da parte principal de segmentação, onde parâmetros como pré-processamento, segmentação e pós-processamento poderão ser ajustados a fim de enviar para a API.

3) *Integração interface e API*: Com a utilização da biblioteca Axios, amplamente utilizada para realizar requisições HTTP, a integração com o back-end foi realizada. Ao fazer uma requisição do tipo POST para o endpoint criado na API, são enviados quatro objetos, além de um *query param*: o arquivo DICOM, um *query param* especificando o método de segmentação, e três JSON: um de pré-processamento, um específico dos parâmetros e segmentação e um de pós-processamento, para filtrar contornos que acabaram sendo reconhecidos pelos métodos. Na resposta da API, é salvo todos os contornos, os contornos válidos e a imagem pré-processada, que é enviada na forma em base 64 pela a API.

IV. CONCLUSÃO

No trabalho apresentado, foi desenvolvido um método de segmentação automática de pulmões em imagens de Tomografia Computadorizada (TC), através do Algoritmo de Contornos Ativos Crisp Adaptivo 2D. O intuito da implementação é reduzir a dependência de segmentações manuais nas imagens, tornando o processo mais eficiente e preciso. Com os resultados obtidos, nota-se que o algoritmo delimita as estruturas pulmonares da forma com que foi solicitada, fornecendo segmentações coerentes e ajustadas às características anatômicas dos pulmões. Vale ressaltar que algumas dificuldades foram encontradas, como a necessidade de ajustes finos em

imagens com variações extremas de contraste e a sensibilidade do algoritmo a determinadas condições de ruído. Com isso, percebe-se que para melhorias futuras pode-se incluir o refinamento dos parâmetros de energia, a incorporação de técnicas de aprendizado de máquina para otimização da segmentação e a integração com sistemas clínicos para aplicações em larga escala. Dito isso, é perceptível que a aplicação do método apresentado contribui para a análise de imagens médicas de forma eficiente, auxiliando na detecção e monitoramento de doenças pulmonares, automatizando as metodologias atuais. Ao reduzir a necessidade de intervir manualmente no processo, o tempo necessário para a análise das imagens diminui e o resultado obtido fica mais preciso, o que torna a ferramenta desenvolvida essencial como apoio ao diagnóstico e à pesquisa na área médica.

REFERÊNCIAS

- [1] Rafael C. Gonzales e Richard E. Woods, "Processamento Digital de 3ed.", Pearson, 2009.
- [2] Aldo von Wangenheim, "Introdução à Visão Computacional".
- [3] UFC. "Algoritmo de Segmentação de Imagens- Algoritmo de Contornos Ativos Crisp Adaptativo 2D Para Segmentação dos Pulmões em Imagens de Tomografia Computadorizada do Tórax, Carlos Victor.
- [4] www.ibm.com, "O que é segmentação de imagem?" disponível em: <https://www.ibm.com/br-pt/topics/image-segmentation>. Acesso em: 15 fev. 2025.

APÊNDICE A CÓDIGO UTILIZADO

Os códigos desenvolvidos no Back-End e Front-End serão disponibilizados em um arquivo separado deste artigo.