



UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ - UNIOESTE

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

SCAPy: Software para determinação de variáveis histopatológicas por Processamento Digital de Imagens em Python

Trabalho de Conclusão de Curso

Vitor Luiz Caldeira Gilnek



Cascavel-PR

2021

UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ - UNIOESTE

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

Vitor Luiz Caldeira Gilnek

**SCAPy: Software para determinação de variáveis
histopatológicas por Processamento Digital de Imagens em
Python**

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel.

Orientador: Claudio Roberto Marquetto Mauricio
Coorientador: Clodis Boscarioli

Cascavel-PR

2021

VITOR LUIZ CALDEIRA GILNEK

**SCAPY: SOFTWARE PARA DETERMINAÇÃO DE VARIÁVEIS
HISTOPATOLÓGICAS POR PROCESSAMENTO DIGITAL DE IMAGENS
EM PYTHON**

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. Claudio Roberto Marquette Mauricio
(Orientador)

Colegiado de Ciência da Computação, UNIOESTE

Prof. Clodis Boscaroli (Coorientador)

Colegiado de Ciência da Computação, UNIOESTE

Profa. Sabrina Grassioli

Colegiado de Ciências Biológicas, UNIOESTE

Cascavel, 27 de julho de 2022

Mesmo com a vastidão do mundo e na imensidão do tempo, você veio parar aqui. Dedico esse texto a você, sim você mesmo que está a ler este trabalho, meu muitíssimo obrigado!

Agradecimentos

Primeiramente, agradeço a toda minha família, especialmente aos meus pais Airton e Lucia pelo apoio e por sempre acreditarem em mim independente de qualquer circunstância.

Às minhas irmãs Amanda e Laura; Amanda por ter sido a melhor irmã mais velha que alguém poderia almejar, sempre me motivando e dando conselhos; e Laura, por ser a melhor irmãzinha que eu poderia desejar ter, sempre alegre e divertida.

Gostaria de agradecer os meu orientadores Cláudio e Clodis e a professora Sabrina por me ajudarem, apoiarem e confiarem em mim neste trabalho, como também queria agradecer a todos os professores do curso de Computação por terem contribuído para a minha formação.

À meu amigo Luís Felipe por sempre proporcionar as melhores conversas dos mais variados assuntos e que, apesar da distância, nossa amizade continuou mais forte do que nunca.

A meus dois grandes amigos Gabriel e Gustavo, em quem eu sempre pude contar com, e que tive diversos momentos memoráveis em jogatinas e conversas.

À meu amigo João Dotta, com quem fiz amizade durante esta jornada e que me acompanhou até o fim, agradeço pelas diversas horas em que conversamos, jogamos e zeramos clássicos do mundo dos vídeo-jogos.

Obrigado Gabi, Val, Pedro, Yuri, Igão e Vedana, por todos os momentos que pude prestigiar junto de vocês.

À todas as pessoas com quem eu tive contato, muito obrigado!

Tudo vale a pena quando a alma não é pequena. (Fernando Pessoa)

Resumo

Através do uso de um microscópio, não apenas é possível obter diversas imagens, as quais seriam impossíveis de visualizar a olho nu, mas também é possível analisar diversos dados que delas podem ser extraídos. Estes dados servem como base para estudos científicos que permitem o desenvolvimento e evolução de tratamentos, diagnósticos de doenças e obtenção de informações pertinentes para estudos futuros. Se tratando de imagens, a computação possui técnicas específicas, chamadas de Processamento Digital de Imagens, que, possibilitam, de maneira otimizada e segura, a extração de dados. Tendo isso em mente, este trabalho tem por objetivo pesquisar e desenvolver um método capaz de contabilizar um determinado número de adipócitos, dada uma imagem retirada de um microscópio, resultando no auxílio dos pesquisadores do Centro de Ciências Biológicas e da Saúde do Campus de Cascavel, de forma a fornecer uma ferramenta que além de rápida e eficiente, seja capaz de poupar o tempo na hora de realizar a quantificação destas células.

Palavras-chave: Processamento Digital de Imagens. Tecido Adiposo. Histologia. Técnicas Digitais.

Lista de figuras

Figura 1 – Diagrama de Tecido Adiposo e Adipócitos	18
Figura 2 – Camundongo obeso (esquerda) comparado a um camundongo normal (direita)	20
Figura 3 – Etapas de um procedimento de PDI	21
Figura 4 – Convenção de eixos usados para representação de imagens digitais.	23
Figura 5 – Diagrama de cromaticidade da CIE	24
Figura 6 – Cubo RGB.	25
Figura 7 – Modelo HSI Triangular e Circular.	26
Figura 8 – Histograma de imagens em escala de cinza	28
Figura 9 – Imagem limiarizada	29
Figura 10 – Imagem com erosão aplicada	30
Figura 11 – Imagem com dilatação aplicada	31
Figura 12 – Imagem com a Transformação da Distância aplicada	32
Figura 13 – Imagem demonstrando os passos da técnica Canny Edge	33
Figura 14 – Imagem demonstrando o <i>Floodfill</i>	34
Figura 15 – Diagrama das Etapas do TCC	39
Figura 16 – Imagem demonstrando desenvolvimento inicial com imagens ruidosas	41
Figura 17 – Imagem demonstrando histograma desenvolvido para a análise da imagem .	42
Figura 18 – Passos do algoritmo sendo executados	43
Figura 19 – Diagrama do fluxo de atividades	44
Figura 20 – Protótipo da Interface	45
Figura 21 – Imagens Selecionadas para a validação	46

Lista de tabelas

Tabela 1 – Comparações de Resultados entre as Imagens	46
---	----

Lista de códigos

Código 1 – Código <i>processing definitions</i>	53
---	----

Lista de abreviaturas e siglas

BAT	<i>Brown Adipocyte Tissue</i>
CCBS	Campus de Cascavel Centro de Ciências Biológicas e da Saúde
CIE	<i>Comission Internationale de l'Eclairage</i>
CWI	<i>Centrum Wiskunde & Informatica</i>
DHGNA	Doença hepática gordurosa não alcoólica
EE	Elemento Estruturante
GUI	<i>Graphical User Interface</i>
H&E	Hematoxilina e Eosina
HSI	<i>Hue, Saturation, Intensity</i>
IMC	Índice de Massa Corporal
MATLAB	<i>Matrix Laboratory</i>
MSC	Glutamato Monossódico
OpenCV	<i>Open Source Computer Vision Library</i>
PDI	Processamento Digital de Imagens
RGB	<i>Red, Blue, Green</i>
SCAPy	<i>Software para Contagem de Adipócitos em Python</i>
WAT	<i>White Adipocyte Tissue</i>

Lista de símbolos

θ	Teta minúsculo
\oplus	Círculo Mais
\ominus	Círculo Menos
\in	Pertence
tan	Tangente
π	Pi
σ	Sigma
atan2	Arcotangente de dois parâmetros

Sumário

1	Introdução	14
1.1	Objetivos Gerais	15
1.2	Objetivos Específicos	15
1.3	Organização dos Capítulos	15
2	Revisão Bibliográfica	17
2.1	Histologia do Tecido Adiposo	17
2.1.1	Adipócitos	18
2.2	Estudos de Tecido Adiposo em Camundongos	19
2.3	Conceitos de Processamento Digital de Imagens	20
2.3.1	História e procedimentos do Processamento digital de Imagens	21
2.3.2	Representação Digital de uma Imagem	22
2.3.3	Modelos de Cores	23
2.3.3.1	Diagrama de cromaticidade da CIE	23
2.3.3.2	RGB	25
2.3.3.3	HSI	25
2.3.4	Histograma	27
2.3.5	Limiarização	28
2.3.6	Erosão de imagens	30
2.3.7	Dilatação de imagem	30
2.3.8	Transformação da Distância	31
2.3.9	Detecção de Bordas de Canny	32
2.3.10	<i>Floodfill</i>	33
2.4	Ferramentas de PDI	34
2.4.1	MATLAB	34
2.4.2	ImageJ	34
2.4.3	OpenCV	35
2.4.4	C++	35
2.4.5	Python	36
2.4.6	Visual Studio Code	36
2.4.7	Qt e Pyside	36
3	Metodologia	37
4	Desenvolvimento e Resultados	40
4.1	Início do desenvolvimento	40

4.2	Implementação de um protótipo	41
4.3	Prototipagem de uma interface	43
4.4	Validação dos Resultados	46
5	Conclusão e Trabalhos Futuros	47
	Referências	49
	Anexos	52
	ANEXO A Código dos passos do Algoritmo desenvolvido	53

1

Introdução

A microscopia é o campo técnico da biologia em que se utilizam microscópios para visualizar e analisar, tanto amostras, quanto objetos, que por sua vez, normalmente não poderiam ser vistos a olho nu (objetos que não estão dentro da faixa de amplitude da resolução de um olho normal) (EDINBURGH, 2018). Em virtude da microscopia, as ciências biológicas e médicas puderam confiar, fortemente, a resolução de problemas relacionados às características morfológicas gerais, bem como servir de ferramenta quantitativa para adquirir e registrar dados específicos. Com isso, o microscópio óptico provou ao longo de sua existência ser útil por nos ajudar a desvendar os inúmeros mistérios da vida (ABRAMOWITZ, 2007).

Neste trabalho, uma das áreas que a microscopia abrange, e iremos abordar, é a área de histologia (estudo dos tecidos), que trata sobre o estudo da anatomia microscópica dos tecidos biológicos a fim de entender a formação, estrutura e função destes. Suas aplicações na Ciência são diversas, tendo como um dos exemplos mais conhecidos a Histopatologia (estudo de doenças através do tecido), na qual é possível extrair diagnósticos e gerar pesquisas sobre doenças. (LEESON; LEESON; PAPARO, 1985) Para podermos extrair essas informações, é feito um processo de obtenção das imagens por meio do escaneamento da lâmina no microscópio. Mas mesmo assim este processo, por muitas vezes, continua sendo manual, sendo não só suscetível a erros como também podendo consumir muito tempo dos pesquisadores que devem analisar as lâminas.

Cada vez mais, o processo de análise histológica, vem sendo auxiliada pela Computação, assim permitindo uma análise que oferta uma precisão alta, com redução de tempo vantajosa. (MCCANN et al., 2015) A área de Processamento Digital de Imagens(PDI), vem se solidificando cada vez mais no campo da microscopia, já que esta possui técnicas que auxiliam a análise semi-assistida e assistida dentro da histologia, sendo possível observar em diversos trabalhos o seu uso, como por exemplo: contagem de cariótipo (KLOCK, 2017), análise de células vermelhas do sangue (MAURICIO, 2017), quantificação de elementos em tecidos. Atualmente, dentro

do Centro de Ciências Biológicas da Unioeste (CCBS), ocorrem diversos tipos de estudos em animais, como por exemplo a comparação de tecido adiposo, levantamento de conjuntos de dados (*datasets*), análise de células e entre outras. Entretanto, todo esse processo ocorre de maneira manual e assistida, podendo haver divergência no apuramento de dados ou obtenção de resultados potencialmente não satisfatório. Tendo em mente todas as informações apresentadas, fica evidente que nossa área tem muito a oferecer, e pode ajudar pesquisadores de outras áreas a mitigar os problemas que estes enfrentam. Busco por meio deste trabalho, desenvolver uma solução e metodologia para a quantificação e parametrização das amostras de tecido adiposo fornecidas pela Profa. Dra. Sabrina Grassioli, utilizando-se das técnicas de PDI para tal.

1.1 Objetivos Gerais

Analisar imagens de secções de tecido adiposo e elaborar uma parametrização e metodologia que permita desenvolver uma ferramenta que atenda os pesquisadores das áreas de ciências biológicas ao executar contagem de adipócitos. Por meio do uso de técnicas de PDI, tem-se por objetivo agilizar os procedimentos, hoje feitos de maneira manual, de análise histológica de adipócitos do CCBS.

1.2 Objetivos Específicos

1. Auxílio do Pesquisador: Fornecer ajuda ao pesquisador do campo da histologia com uma ferramenta confiável e que lhe poupe tempo ao analisar amostras de tecido adiposo.
2. Criação de *datasets*: Criar um conjunto de imagens microscópicas de adipócitos para estudos e desenvolvimento da ferramenta, para a realização de testes e obtenção de resultados.
3. Desenvolvimento de Parâmetros: Desenvolver parâmetros e tutoriais para padronização dos procedimentos de análise histológica dentro do CCBS com o intuito de se obter um melhor resultado ao executar o sistema.

1.3 Organização dos Capítulos

Este documento segue organizado da seguinte forma:

- No Capítulo 2, são tratados os assuntos necessários para o entendimento do problema, onde é demonstrada uma visão geral do tecido adiposo, sua constituição e problemas que derivam dele, assim como também é abordado conceitos sobre as técnicas e ferramentas de processamento digital de imagens que auxiliarão no desenvolvimento do sistema responsável por realizar a análise e contagem dos adipócitos da imagem.

-
- No Capítulo 3 é apresentada a metodologia que foi utilizada no desenvolvimento da solução para o problema, salientando as fases do projeto a que foram seguidas assim como uma amostra das ferramentas e tecnologias estudadas para a utilização no projeto.
 - No Capítulo 4 é abordada os resultados que foram alcançados durante a execução desta monografia, como a implementação e validação do sistema e uma discussão que reúne os principais pontos sobre o projeto.
 - No Capítulo 5 é exposto as conclusões após obter-se os resultados apresentados anteriormente, onde nesta é discutido possíveis trabalhos futuros utilizando este projeto como base ou inspiração

2

Revisão Bibliográfica

Neste capítulo será apresentado uma visão geral dos conceitos que serão necessários para entender tanto o problema, quanto para ajudar a desenvolver sua solução.

2.1 Histologia do Tecido Adiposo

O tecido adiposo nada mais é do que uma variedade do tecido conjuntivo formada predominantemente por adipócitos. Ele é principalmente localizado embaixo da pele (gordura subcutânea), ao redor dos órgãos internos (gordura visceral), na medula óssea e outros, onde estes são chamados de depósitos de adipócitos. Possui diversas funções, podendo servir como isolamento térmico do organismo como também depósito de energia para o corpo (JUNQUEIRA; CARNEIRO, 1999). O tecido adiposo possui dois tipos, que podem ser divididos da seguinte forma:

Tecido Adiposo Branco: *White Adipocyte Tissue* (Tecido Adiposo Branco) ou WAT é composto de adipócitos monoculares, na qual estes contêm uma única gota grande de gordura, o que faz com que o núcleo passe a ser espremido para a borda da célula. Elas possuem receptores para insulina, hormônios sexuais, norepinefrina e glicocorticoides. (REYES-FARIAS et al., 2021)

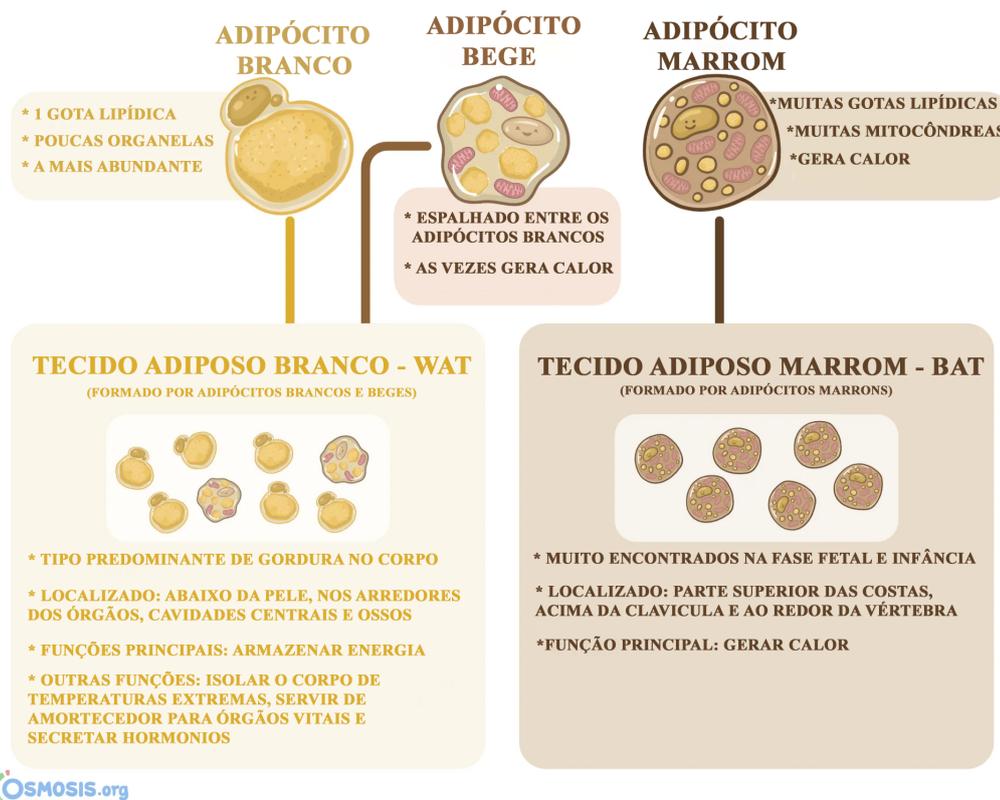
Tecido Adiposo Marrom: Já o *Brown Adipose Tissue* (tecido adiposo marrom) ou BAT refere-se a duas populações celulares distintas com funções semelhantes. A primeira destas compartilha uma origem embriológica em comum com as células musculares, em que são encontradas em depósitos "clássicos" maiores. A outra se desenvolve a partir de adipócitos brancos que são estimulados pelo sistema nervoso simpático. Estes adipócitos são encontrados intercalados com tecido adiposo branco e são chamados de 'bege' ou 'brite' (marrom em branco).

Sua função primária é a termorregulação, na qual ele produz calor pela termogênese. Em contrapartida, ao analisar o BAT com o WAT, que contém uma única gotícula lipídica, os BATs contêm numerosas gotículas menores e um número muito maior de mitocôndrias (contendo ferro), o que dá cor ao tecido. (CINTI, 2001)

2.1.1 Adipócitos

As células adiposas, conhecida também por lipócitos ou adipócitos são as principais células de composição do tecido adiposo. Elas são formadas por um processo chamado adipogênese, em que uma célula tronco mesenquimal multipotente origina um lipoblasto, e este acumula gotículas de gordura em seu citoplasma. (GRAAFF, 2003) Como citado anteriormente, os adipócitos podem ser divididos em tipos e funções. Abaixo na figura 1 é possível visualizar um diagrama que demonstra visualmente as diferenças entre os adipócitos brancos e marrons, e os tecidos, que salienta suas principais características e localização dentro do corpo.

Figura 1 – Diagrama de Tecido Adiposo e Adipócitos



Fonte: Adaptado de Hernández (2021)

2.2 Estudos de Tecido Adiposo em Camundongos

A obesidade é uma condição definida como um acúmulo excessivo ou anormal de tecido adiposo que pode impactar negativamente a saúde da pessoa. O Índice de Massa Corporal (IMC) é um índice que faz o cálculo de peso por altura, e é utilizado para identificar e classificar sobrepeso e obesidade. Uma pessoa pode ser considerada obesa quando seu IMC é superior a 30 kg/m². Ela, por sua vez, aumenta consideravelmente a chance da pessoa contrair certas doenças, como diabetes tipo 2, apneia, osteoartrite e doenças cardiovasculares. ([World Health Organization, 2015](#))

Tanto a obesidade quanto o envelhecimento tem ligações com doenças metabólicas, como doenças cardiovasculares e diabetes tipo 2 que haviam sido citadas anteriormente. A inflamação crônica do tecido adiposo é um dos mecanismos que acarretam na progressão dessas doenças, já que elas desencadeiam alterações no tecido adiposo que levam ao desenvolvimento de células imunes com fenótipo pró-inflamatório. Um estado inflamatório sistêmico crônico, promove também o desenvolvimento de muitas doenças, como síndrome metabólica, doenças cardiovasculares, doença hepática gordurosa não alcoólica (DHGNA) e outras. ([REYES-FARIAS et al., 2021](#))

Diversos trabalhos sobre estudos de WAT e BAT são realizados em camundongos, visto que existe uma similaridade genética estrutural muito grande comparados à estrutura genética humana, fazendo deles uma plataforma confiável para pesquisa sobre doenças como problemas metabólicos, defeitos neurológicos, obesidade e muitos outros. ([KUDA; ROSSMEISL; KOPECKY, 2018](#))

Para obter-se os dados relativos aos tecidos adiposos, existem diversas formas de fazer o camundongo desenvolver adipócitos, sejam eles BAT ou WAT. Temos como exemplo o trabalho de [Wang et al. \(2013\)](#) onde durante o desenvolvimento, a alimentação do camundongo é feita com uma rica dieta em gordura chamada '*High Fat Diet*' (HFD) e exposição ao frio. Com isso há o "escurecimento" da gordura subcutânea, que é de onde a maioria dos adipócitos "beges" derivam. Durante a alimentação HFD, a gordura epididimal começa a sofrer adipogênese após 4 semanas, enquanto na gordura subcutânea, esta sofre hipertrofia por um período de até 12 semanas.

Dentre outras maneiras que é possível gerar amostras de WATs, temos os camundongos obesos. Esses camundongos carecem de leptina funcional e desenvolvem muita resistência a insulina, assim sendo, eles têm sido muito usados como modelo para obesidade. A deficiência de leptina manifesta-se também em relação à função imune, ao sistema cardiovascular incluindo angiogênese, função do tecido de suporte, malignidades e função reprodutiva. ([LINDSTRÖM, 2007](#))

Sem a leptina, o camundongo fica muito mais faminto do que um camundongo normal, levando ele a comer muito mais e acarretando em sua obesidade, na qual ele possuirá um depósito de tecido adiposo muito maior, como é possível ver na imagem [2](#).

Dentre os diversos modelos citados, um dos utilizados dentro da universidade adaptado de (LUBACZEUSKI et al., 2015) consiste criar pequenos ratos, que quando recém-nascidos recebem glutamato monossódico (MSG), aplicando 4 g/kg de peso corporal, ou solução salina durante os primeiros 5 dias de vida, e depois aos 30 dias de idade, ambos os grupos de ratos são submetidos à cirurgia simulada ou vagotomia subdiafragmática, fazendo com que eles apresentem obesidade, hiperinsulinemia, resistência à insulina e hipertrigliceridemia.

Figura 2 – Camundongo obeso (esquerda) comparado a um camundongo normal (direita)



Fonte: (Oak Ridge National Laboratory, 1999)

Para analisarmos o WAT ou BAT de um camundongo se faz necessário o uso de corantes para a visualização destes após a extração do tecido. Ao fazer a secção desses tecidos é proposto por Cardiff, Miller e Munn (2014) o uso de hematoxilina e eosina (H&E), que é o padrão usado para exame microscópico de tecidos que foram fixados, processados, incorporados e seccionados. Este protocolo descreve a coloração manual de H&E de tecidos de camundongos fixados, processados, embebidos em parafina e seccionados. Em tecidos corados com H&E, os ácidos nucleicos coram em azul escuro e as proteínas coram de vermelho a rosa ou laranja. Com uma cor apropriada e uma boa coloração é possível identificar em diversos tecidos detalhes valiosos que, combinados com técnicas de processamento digital de imagens, fornecem diagnósticos precisos.

2.3 Conceitos de Processamento Digital de Imagens

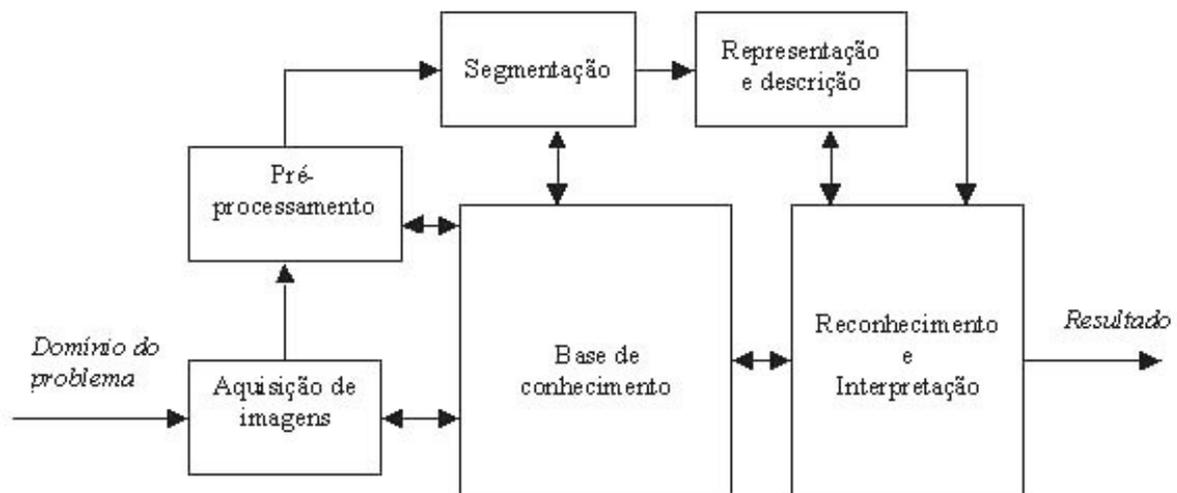
Nesta seção, para fins de melhor entendimento ao leitor, foram inseridas imagens de adipócitos fornecidas pelo CCBS da UNIOESTE (Universidade Estadual do Oeste do Paraná), onde nestas, foram aplicadas algumas das técnicas descritas no capítulo da revisão Bibliográfica.

2.3.1 História e procedimentos do Processamento digital de Imagens

Os primeiros computadores a serem capazes de realizar importantes tarefas de processamento digital de imagens, foram desenvolvidos no início da década de 1960 e a partir desta, a área de processamento digital de imagens cresceu rapidamente. Assim sendo, as técnicas de Processamento Digital de Imagens (PDI), são ferramentas muito úteis que nos auxiliam nas mais diversas tarefas, como por exemplo identificação facial, restauração de imagens e da arte, automação da indústria, e no caso deste trabalho, análise de imagens microscópicas. (GONZALEZ; WOODS, 2018)

Podemos reduzir o procedimento de processamento de imagens em 5 etapas, como é apresentado na figura abaixo, com uma descrição sucinta de cada uma destes passos.

Figura 3 – Etapas de um procedimento de PDI



Fonte: Gonzalez e Woods (2002)

Aquisição de imagens: Nesta etapa é realizada a captura da imagem a ser processada, seja ela por microscópio, scanner, câmera ou outro meio, ela sofre um processo de digitalização para um formato que seja possível de ser processado em ambiente digital.

Pré-processamento: Nem toda imagem que passa pelo processo de digitalização possui o fortúnio de sair perfeita, já que em sua grande maioria, elas podem apresentar ruídos, problemas de contraste e/ou iluminação. Entretanto esta etapa tem como objetivo corrigir estes empecilhos, utilizando-se de técnicas que permitem ajustar a imagem ao problema a ser solucionado.

Segmentação: Esta parte fica responsável por isolar ou ressaltar características, propriedades ou até áreas da imagem com o intuito de adquirir alguma informação pertinente ao problema.

Consiste de estratégias que detectam similaridades dentro da imagem, podendo aumentar muito as chances de solução do problema e auxiliando nas próximas etapas.

Representação e Descrição: Neste passo, é utilizado os dados da etapa de segmentação para uso de para verificação de padrões de representação, e depois é extraído e armazenado as informações que possuem os atributos para a análise da imagem.

Reconhecimento e Interpretação: E por último mas não menos importante, é atribuído um rótulo para identificá-lo respaldado em sua representação, que fora adquirida no passo anterior. E assim é interpretada e definido um significado aos objetos analisados da imagem para a solução do problema.

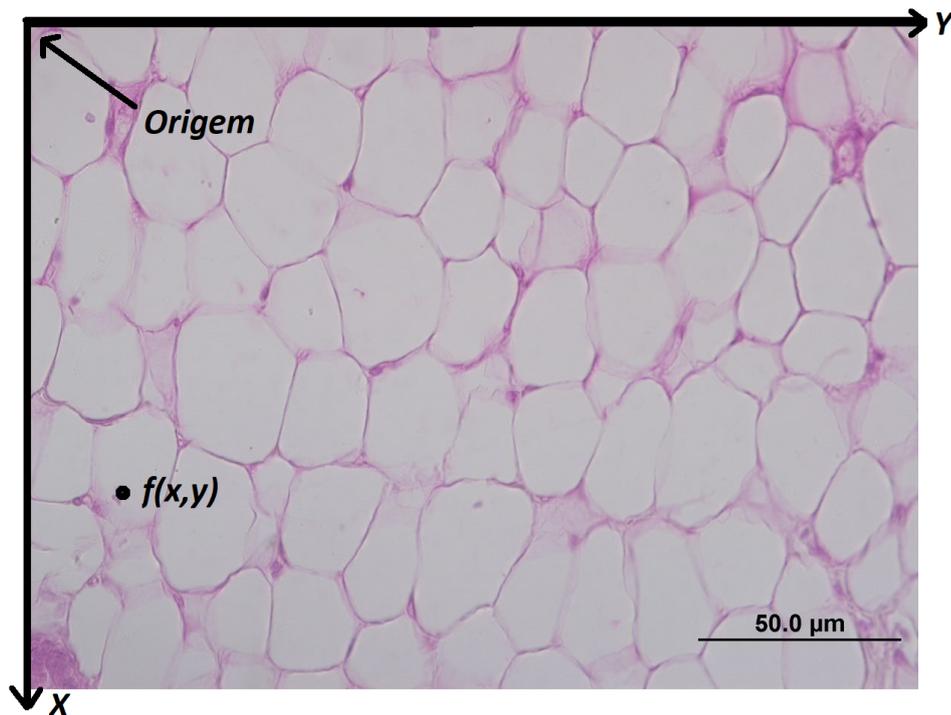
2.3.2 Representação Digital de uma Imagem

Uma representação de uma imagem pode ser descrita como uma função bidimensional $f(x, y)$, tal que x e y são coordenadas espaciais e o valor de f em qualquer ponto (x, y) é proporcional à intensidade da cor da imagem naquele ponto. Em uma imagem monocromática, cada ponto (x, y) , é dada por uma função f da intensidade da luz, sendo que o valor de f é proporcional ao brilho da imagem naquele ponto.

Quando tratamos de uma imagem digital bidimensional, nos referimos a uma imagem $f(x, y)$ que foi discretizada em suas coordenadas espaciais e no brilho. Ou seja, quando uma imagem possui valores de intensidade de f , x e y finitas e discretas, podemos chamar esta de imagem digital.

Uma imagem digital pode ser considerada como uma matriz M cujos índices de linhas e colunas, i e j respectivamente, identificam pontos na imagem e o valor de um elemento $(M(i, j))$ da matriz é o valor de um ponto $(f(x, y))$ na imagem. Esses elementos são chamados de elementos de imagem, pels e pixels. O termo pixel é o mais utilizado para denotar os elementos de uma imagem digital. (GONZALEZ; WOODS, 2018) Podemos observar esta convenção de eixos através da imagem 4.

Figura 4 – Convenção de eixos usados para representação de imagens digitais.



Fonte: Imagem fornecida pela professora Sabrina Grassioli, modificado pelo Autor

2.3.3 Modelos de Cores

Embora o processo empregado pelo cérebro humano na percepção e interpretação da cor seja um fenômeno fisiopsicológico que ainda não é totalmente compreendido, a natureza física da cor é empírica. Basicamente, as cores que os humanos e alguns outros animais percebem em um objeto são determinadas pela natureza da luz refletida pelo objeto. Para facilitar e padronizar os métodos de visualização de cores, foram criados modelos de cores. Essencialmente, um modelo, ou espaço de cor é uma especificação de um sistema de coordenadas tridimensionais e um subespaço dentro deste sistema, onde cada cor é representada por um único ponto. (GONZALEZ; WOODS, 2018) Em sua grande maioria, os modelos de cores são mais orientados para hardware, e dentre a grande diversidade destes, os que são mais empregados dentro do campo de PID, são o diagrama de cromaticidade da CIE, RGB e HSI que serão descritas a seguir.

2.3.3.1 Diagrama de cromaticidade da CIE

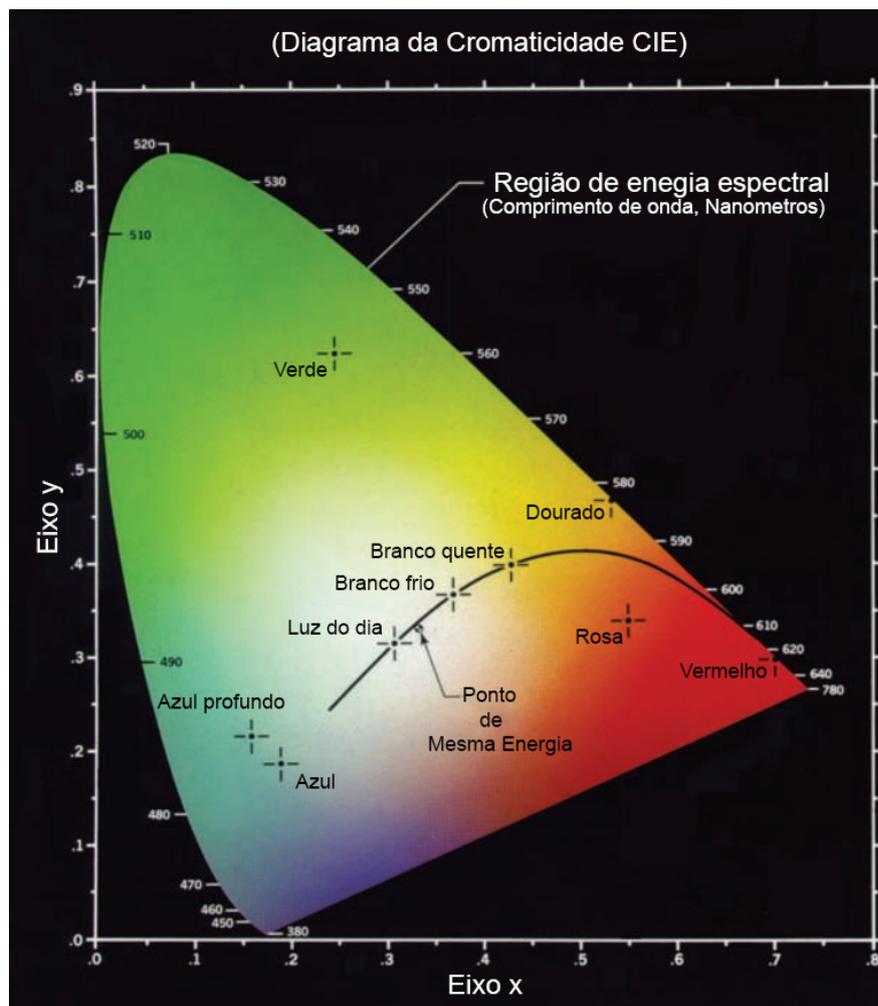
CIE, do francês *Commission Internationale de l'Eclairage* (Comissão Internacional de Iluminação), é uma organização que desde sua criação, tem como propósito padronizar as cores através de seus comprimentos de ondas. Através de muita experimentação e estudos, a CIE criou um diagrama de cromaticidade, que mostra a composição de cores em função de x (vermelho) e y (verde). As posições das várias cores do espectro - de violeta em 380 nm a vermelho em 780 nm - são indicados ao redor do limite do diagrama de cromaticidade em forma de língua. Qualquer

ponto que não esteja no limite, mas esteja dentro do diagrama, representa alguma mistura das cores puras do espectro. O ponto de energia igual mostrado na 5 corresponde a frações iguais das três cores primárias, representa o padrão CIE para branco, e qualquer ponto localizado no limite do gráfico de cromaticidade está totalmente saturado.

À medida que um ponto deixa o limite e se aproxima do ponto de igual energia, mais luz branca é adicionada à cor e ela se torna menos saturada, já que a saturação do ponto de igual energia é zero. Se houver mais luz vermelha do que verde, o ponto exato que representa essa nova cor estará na segmento de reta, entretanto estará mais próximo do ponto vermelho do que do verde. Da mesma forma, uma linha traçada do ponto de igual energia a qualquer ponto no limite do gráfico definirá todos os tons desse espectro de cor.

O diagrama de cromaticidade é útil para a mistura de cores porque um segmento de linha reta que une dois pontos quaisquer no diagrama define todas as diferentes variações de cores que pode ser obtido combinando essas duas cores aditivamente. Vamos considerar, por exemplo, uma linha reta traçada do ponto vermelho ao ponto verde mostrados na Figura 5.

Figura 5 – Diagrama de cromaticidade da CIE

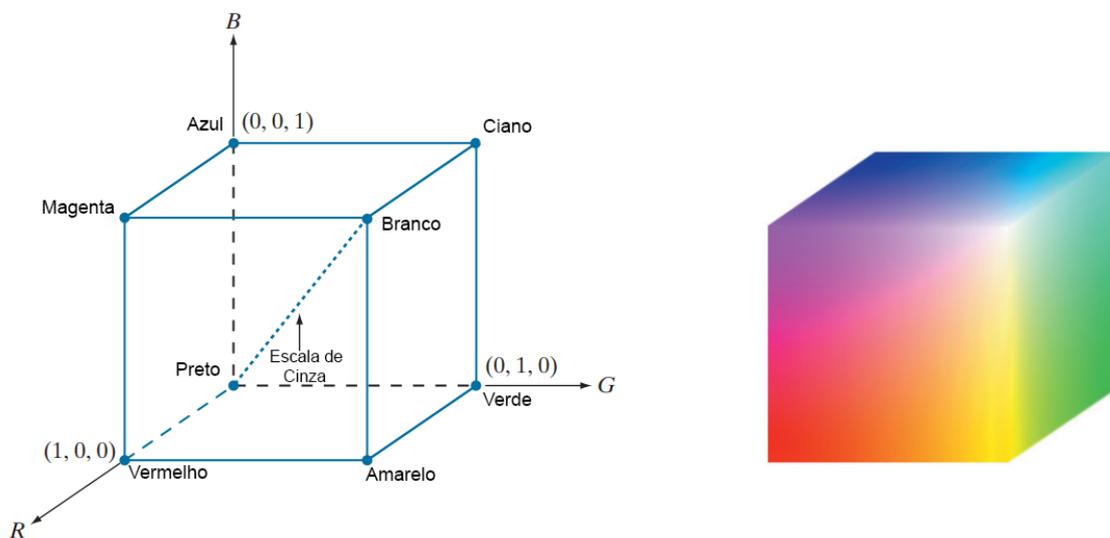


Fonte: Adaptado de [Gonzalez e Woods \(2018\)](#)

2.3.3.2 RGB

RGB, do inglês, *Red* (Vermelho), *Green* (Verde) e *Blue* (Azul), é um modelo de cor, que é descrito através de um sistema de coordenadas cartesiano, onde os vértices são representados tanto pelas cores primárias complementares (ciano, magenta e amarelo) quanto pelas cores primárias padrão (vermelho, verde, azul). (GONZALEZ; WOODS, 2018) Já a diagonal principal possui valores de cinza, onde na origem é preto, e no final do vértice é branco. É possível visualizar a representação deste plano na figura a seguir.

Figura 6 – Cubo RGB.



Fonte: Adaptado de [Gonzalez e Woods \(2018\)](#)

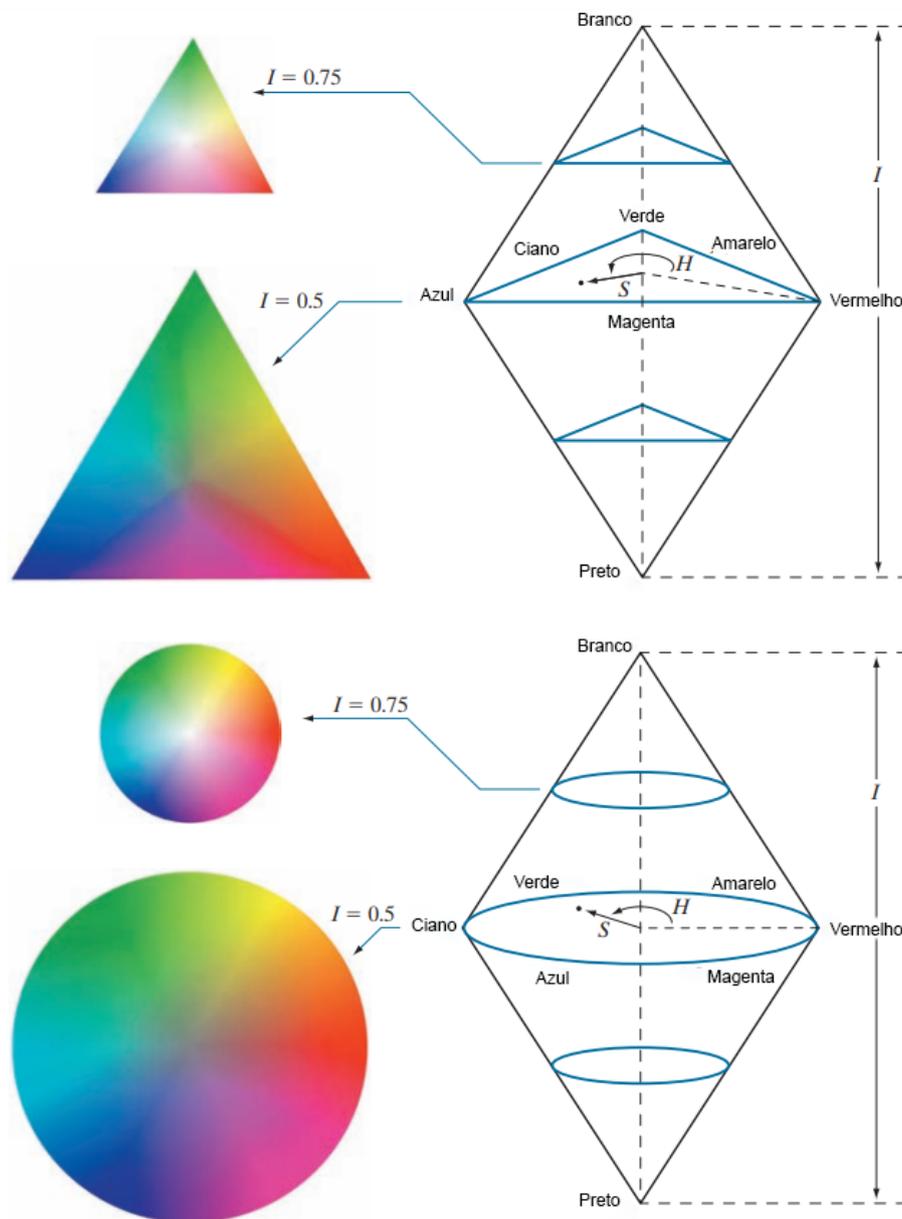
Resumidamente, as imagens no modelo RGB consistem em três planos de imagens independentes, uma para cada cor primária, onde, quando combinadas, formam a imagem colorida. Em um monitor ou televisão por exemplo, estes três planos se combinam para produzir uma cor composta dentro do espectro do modelo RGB. Observando a figura fica perceptível o gradiente de cores proporcionado por esse modelo, assim, o uso do modelo RGB para processamento de imagens, faz sentido quando as imagens são expressas nestes três planos de cores. Com isso podemos afimar o grau de importância que o uso do RGB dentro do campo processamento de imagens.

2.3.3.3 HSI

O modelo que é tratado nesta seção, é chamado de modelo de cor HSI, do inglês *Hue* (matiz), *Saturation* (saturação), *Intensity* (intensidade), desacopla o componente de intensidade das informações de transporte de cor (matiz e saturação) em uma imagem colorida. Assim temos que, o modelo HSI é uma ferramenta útil para desenvolver algoritmos de processamento de

imagem baseados em descrições de cores que são naturais e intuitivas para nós seres humanos, que, no final, acabam sendo os usuários e desenvolvedores destes algoritmos. (GONZALEZ; WOODS, 2018) Na figura 7 é possível ver a representação deste modelo, tanto por um planos de cores triangulares quanto (b) circulares. Os triângulos e círculos são perpendiculares ao eixo vertical de intensidade. Neles temos que o ponto é qualquer ponto de cor, o ângulo do eixo vermelho nos fornece a matiz, o comprimento de o vetor é a saturação e a intensidade de todas as cores em qualquer um destes planos é dado pela a posição do plano no eixo de intensidade vertical.

Figura 7 – Modelo HSI Triangular e Circular.



Fonte: Adaptado de Gonzalez e Woods (2018)

Resumidamente, podemos afirmar que o RGB é ideal para geração de cores de imagens

(como na captura de imagens por uma câmera colorida ou exibição de imagens em uma tela de monitor), entretanto seu uso para descrição de cores é muito mais limitado. Neste caso é possível realizar uma conversão do modelo RGB para o HSI utilizando-se das seguintes fórmulas:

Para o cálculo do *Hue*.

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases} \quad (1)$$

com,

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{\frac{1}{2}}} \right\} \quad (2)$$

Para o cálculo da *Saturation*.

$$S = 1 - \frac{3}{(R + G + B)} [\min(R, G, B)] \quad (3)$$

E por último, para o cálculo da *Intensity*.

$$I = \frac{1}{3}(R + G + B) \quad (4)$$

2.3.4 Histograma

O histograma é uma distribuição de frequências dos níveis de intensidade das cores de uma imagem de acordo com seu tom, tal que no gráfico as tonalidades mais puxadas para a cor preta ficam a esquerda, as com tonalidades brancas a direita, e no meio cinza. Para normalizar, é usada a seguinte formula

$$h(r_k) = n_k / MN \quad (5)$$

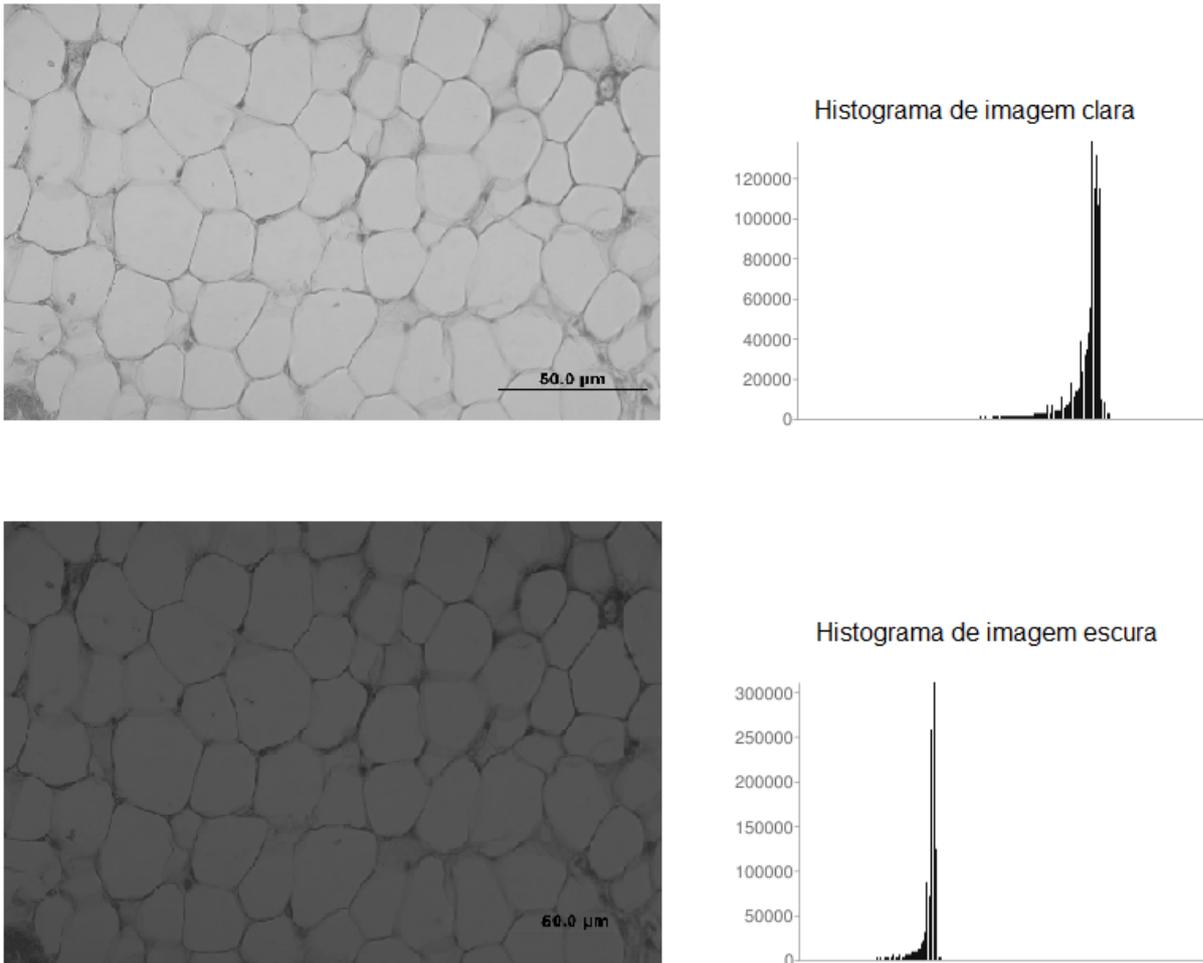
Assim temos que M e N são as linhas e colunas (respectivamente) da imagem para $k = 0, 1, 2, 3, \dots, L-1$, e, $h(r_k)$ a probabilidade de ocorrência daquele nível de cor, tal que a soma de todos os componentes da normalização se iguale a 1. Ao normalizar, possuem um intervalo de 0 a L-1, que é obtida pela função discreta abaixo:

$$h(r_k) = n_k \quad (6)$$

É possível aferir que n_k representa o número de pixels com aquela intensidade, e r_k o k -ésimo valor de intensidade. (GONZALEZ; WOODS, 2018)

Abaixo é possível conferir algumas imagens na figura 8 que exemplificam o funcionamento do histograma quando aplicado em imagens já transformadas em escala de cinza.

Figura 8 – Histograma de imagens em escala de cinza



Fonte: Autor

Com as informações obtidas do histograma, podemos utilizar-se delas para avaliar contrastes da imagem, manipular esta para ressaltar áreas específicas e obter estatísticas com estes resultados.

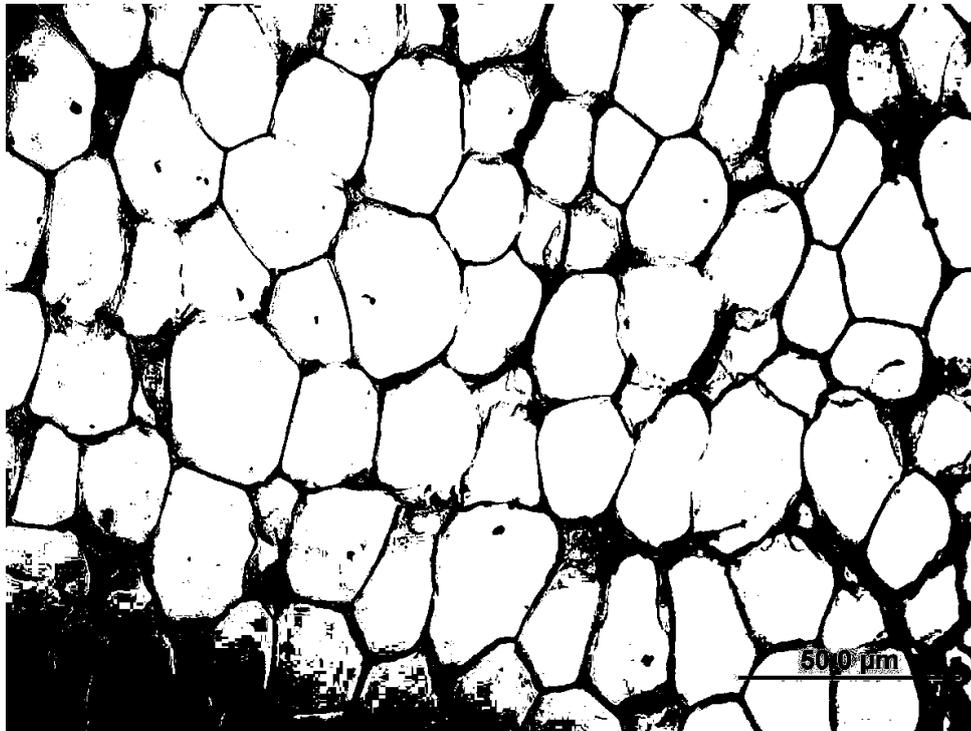
2.3.5 Limiarização

Limiarização ou *thresholding*, é a técnica que permite que você segmente objetos de uma imagem através do uso de um limiar T que divide a intensidade das cores da imagem em grupos. (GONZALEZ; WOODS, 2018) A função $\{g(x,y)\}$ a seguir demonstra como é esta divisão dos grupos:

$$g(x,y) = \begin{cases} 0, & \text{if } f(x,y) \leq T \\ 1, & \text{if } f(x,y) > T \end{cases} \quad (7)$$

Através dessa divisão binária de grupos temos valores de intensidade de 0 e 1, onde os rotulados 0 são considerados pontos do fundo e os que são rotulados como 1 são considerados pontos do objeto. É possível visualizar a aplicação desta técnica na figura 9

Figura 9 – Imagem limiarizada



Fonte: Autor

A limiarização abre possibilidade para diferentes tipos ao modificarmos a constante de limiar, como por exemplo limiarização global (valor constante do limiar T), variável (valor variável do limiar T), local (valor do limiar T dependente das propriedades dos píxeis vizinhos) e dinâmica (valor do limiar T dependente das coordenadas $\{(x,y)\}$). (GONZALEZ; WOODS, 2018)

Também existe a possibilidade do uso de mais de um limiar T , a qual chamamos de limiarização múltipla. Esta pode ser expressa pela seguinte equação:

$$g(x,y) = \begin{cases} a, & \text{if } f(x,y) > T_2 \\ b, & \text{if } T_1 < f(x,y) < T_2 \\ c, & \text{if } f(x,y) \leq T_1 \end{cases} \quad (8)$$

Esse tipo de limiarização é muito utilizada geralmente quando a imagem possui ruídos, objetos com tons variados, iluminação uniforme e diversos outros fatores. (PEDRINI; SCHWARTZ, 2008)

2.3.6 Erosão de imagens

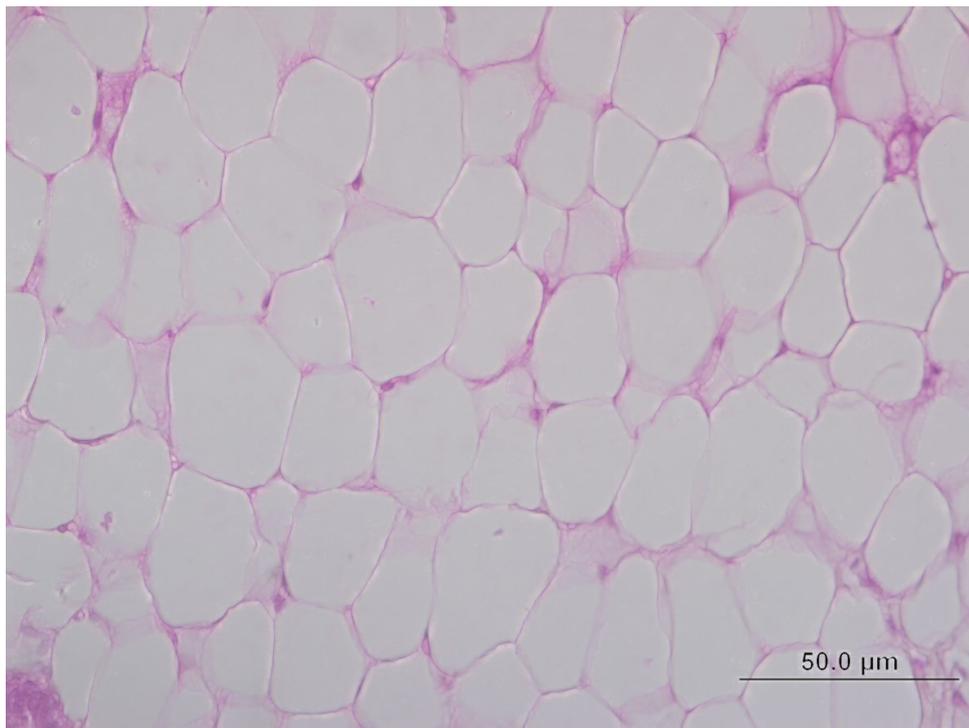
A erosão é uma técnica de filtro morfológico que desloca um elemento estruturante (EE) sobre uma imagem, encontrando o valor mínimo dela na sub-região gerada por essa sobreposição do EE na imagem. Devido as diversas formas de um EE, sua aplicação pode prover diferentes resultados. (GONZALEZ; WOODS, 2018)

A técnica de erosão pode ser matematicamente representada por \ominus em qualquer ponto (x,y) em uma imagem f por um EE pela seguinte equação:

$$[f \ominus b](x, y) = \min_{(s,t) \in b} \{f(x + s, y + t)\} \quad (9)$$

Através disto, é possível perceber que a operação da técnica de erosão calcula o valor mínimo em cada sub-região, e, a imagem resultante na maioria dos casos acaba se tornando mais escura que a original, pois o tamanho das regiões escuras é aumentado e das regiões claras é reduzido. Um exemplo das alterações geradas pela erosão pode ser visto na 10.

Figura 10 – Imagem com erosão aplicada



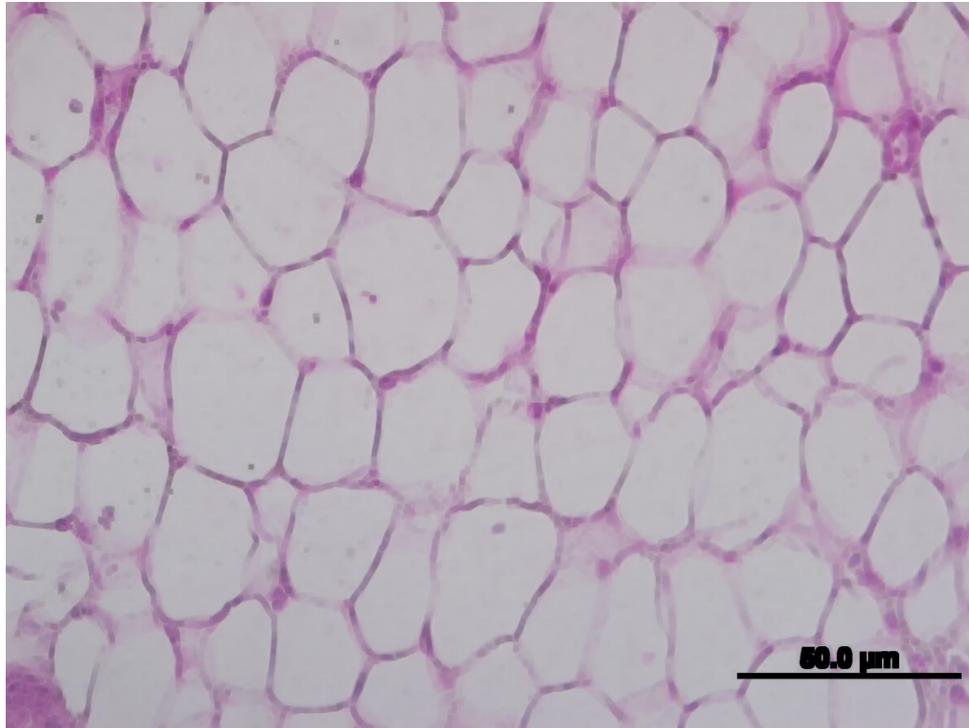
Fonte: Autor

2.3.7 Dilatação de imagem

A dilatação, assim como a erosão, é uma técnica de filtro morfológico, entretanto seu funcionamento difere da anterior. Nesta em vez de calcular o valor mínimo da imagem na sub-região, ela calcula o valor máximo. Quando é calculado este valor, é esperado que o resultado

reduza o tamanho das áreas escuras e aumente o valor das áreas claras, como é possível ver na 11. (GONZALEZ; WOODS, 2018)

Figura 11 – Imagem com dilatação aplicada



Fonte: Autor

A técnica de dilatação pode ser matematicamente representada por \oplus em qualquer ponto (x,y) em uma imagem f por um EE pela seguinte equação:

$$[f \oplus b](x, y) = \max_{(s,t) \in b} \{f(x - s, y - t)\} \quad (10)$$

Tal que s e t variam dependendo das dimensões da matriz que contem o EE.

2.3.8 Transformação da Distância

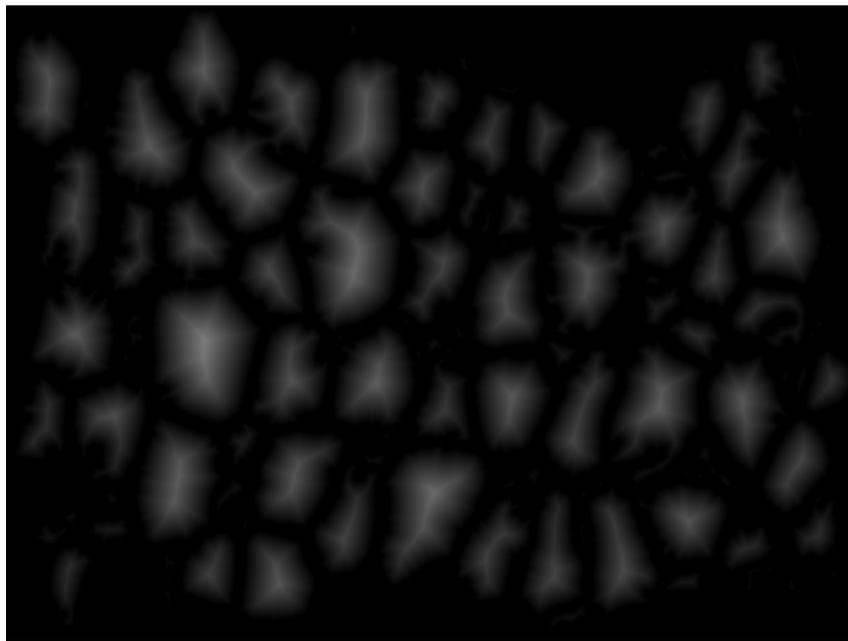
Uma transformação de distância processa uma imagem digital binarizada, onde, consistindo de pixels com características e sem características, todos os pixels sem características têm um valor correspondente a a distância até o pixel de recurso mais próximo.(BORGEFORS, 1986)

Uma maneira mais intuitiva de se pensar sobre a transformação de distância, consiste em imaginar que as regiões de primeiro plano na imagem binária de entrada são feitas de algum material inflamável uniforme de queima lenta e que em seguida, considere um incêndio em todos os pontos no limite de uma região de primeiro plano onde o fogo irá queimar seu caminho para o interior. Se então rotularmos cada ponto no interior com a quantidade de tempo que o fogo levou

para chegar àquele ponto, então efetivamente computamos a transformação de distância daquela região. ([Robert Fisher and Simon Perkins and Ashley Walker and Erik Wolfart, 2003](#))

Na figura 12 consegue se observar essa ideia anteriormente citada, tendo como resultado uma imagem de nível de cinza, exceto que as intensidades de nível de cinza dos pontos dentro das regiões de primeiro plano são alteradas para mostrar a distância até o limite mais próximo de cada ponto.

Figura 12 – Imagem com a Transformação da Distância aplicada



Fonte: Autor

2.3.9 Detecção de Bordas de Canny

Esta técnica que foi desenvolvida por Jhon F. Canny, utiliza um algoritmo multi-estágios com o intuito de detectar diversos tipos de limiares e bordas dentro de uma determinada imagem. ([CANNY, 1986](#))

O primeiro estágio desse algoritmo visa diminuir os ruídos da imagem, já que estes afetam diretamente na detecção das bordas, podendo criar detecções falsas. Imaginando um kernel de tamanho $(2k + 1)(2k + 1)$, esta operação pode ser representada da seguinte forma:

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i - (k + 1))^2 + (j - (k + 1))^2}{2\sigma^2}\right); 1 \leq i, j \leq (2k + 1) \quad (11)$$

Após a aplicação do filtro, o próximo passo é encontrar o gradiente de intensidade da imagem, já que a borda de uma imagem pode variar de direção. Para isso esta parte do algoritmo utiliza 4 filtros para detecção vertical, horizontal e diagonal nas bordas da imagem tratada. Isto pode ser definido através da equação:

$$G = \sqrt{G_x^2 + G_y^2} \quad (12)$$

$$\Theta = \text{atan2}(G_y, G_x) \quad (13)$$

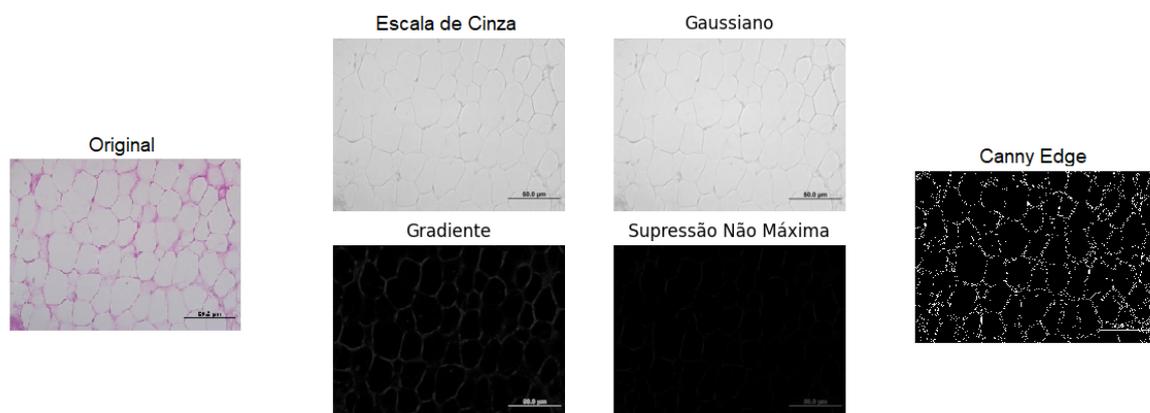
Esta operação retorna um valor derivado na direção horizontal (G_x) e na direção vertical (G_y), onde G pode ser contado utilizando-se da função da hipotenusa e da arcotangente dos dois arumentos.

O terceiro passo do algoritmo consiste em aplicar uma supressão não máxima, de forma com que se possa afinar as bordas, entretanto esta gera algumas variações onde algum pixel da borda aparenta ter mais brilho do que na original. Estas que serão tratadas nos próximos passos.

Os últimos passos são aplicar um duplo threshold, que visa identificar pixels fracos fortes e não relevantes para a detecção de borda, e rastreamento de borda por histerese, que irá utilizar o resultado adquirido do passo anterior para transformar os pixels fracos em fortes, caso exista um pixel adjacente que seja um pixel forte, assim concluindo o processo desta técnica.

Na imagem 13 é possível ver todos estes passos sendo executados desde a imagem inicial até a final.

Figura 13 – Imagem demosntrando os passos da técnica Canny Edge



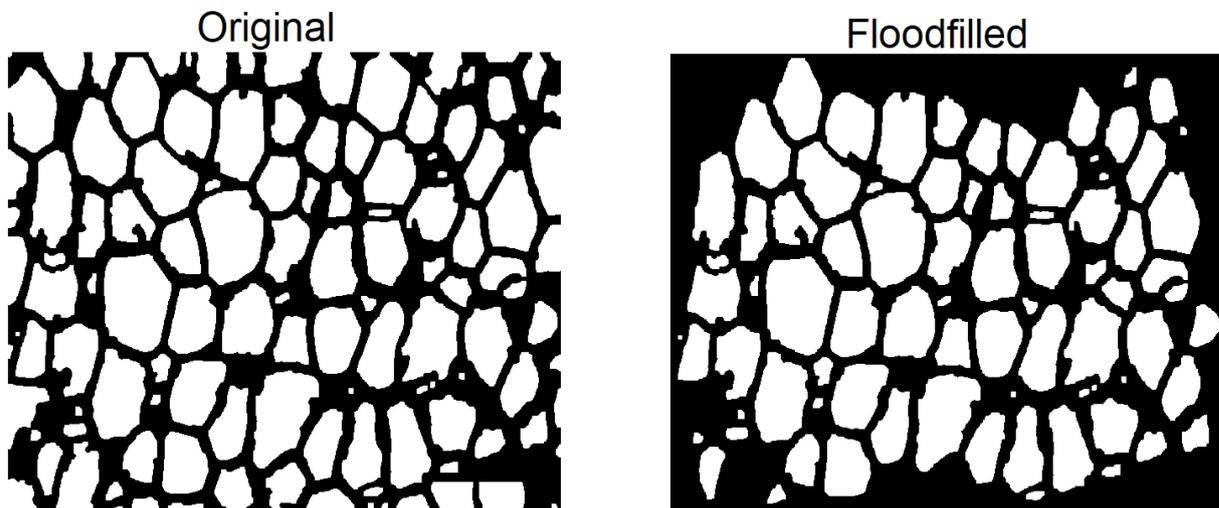
Fonte: Autor

2.3.10 *Floodfill*

Floodfill, é um algoritmo de preenchimento de pixels, que determina e altera a área conectada a um determinado nó em uma matriz multidimensional com um atributo estabelecido correspondente. Esta técnica é utilizada por exemplo, na ferramenta de preenchimento de cor "balde" de programas de pintura, onde ao utilizar, ela preenche áreas conectadas de cor semelhante

com uma cor diferente. (SIGGRAPH. . . , 1979) A imagem 14 demonstra como esse procedimento ocorre, onde utilizando o algoritmo e criando uma borda de 1 pixel da cor branca, ele preenche todos os objetos brancos que compartilham a borda.

Figura 14 – Imagem demonstrando o *Floodfill*



Fonte: Autor

2.4 Ferramentas de PDI

Nesta seção serão descritos linguagens de programação e ferramentas que auxiliam o desenvolvimento de soluções de PDI.

2.4.1 MATLAB

Matrix Laboratory ou mais comumente conhecido como MATLAB, é um software de cálculo e manipulação de matrizes, utilizado por milhões de engenheiros e cientistas para analisar dados, desenvolver algoritmos e criar modelos. Com ele suas aplicações podem ir desde programação paralela até *deep learning* (MATHWORKS, 2022). Este programa provê essas diversas aplicações através das chamadas *toolboxes* (caixas de ferramentas), onde com elas você pode aplicar algoritmos e criar aplicativos para resolver um problema. No caso de PDI existe uma *toolbox* específica chamada *Image Processing Toolbox*, que possui diversos filtros, como por exemplo os filtros citados anteriormente (limiarização, dilatação, erosão, histograma) e diversos outros que vão desde redução de ruídos e filtros até transformações geométricas.

2.4.2 ImageJ

ImageJ é um programa de processamento de imagem Java de domínio público criado por Wayne Rasband em 1997, que tem inspiração direta no NIH Image para Macintosh. Com ele é

possível além de diversas outras funcionalidades: (RASBAND, 2022)

- Exibir, editar, analisar, processar, salvar e imprimir imagens de 8 bits, 16 bits e 32 bits.
- Ler diversos formatos de imagem, incluindo TIFF, GIF, JPEG, BMP, DICOM, FITS e *raw* com suporte a *stacks*, que são uma série de imagens que compartilham uma única janela.
- Executar em multithread, de modo que operações demoradas, como leitura de arquivos de imagem, podem ser executadas em paralelo com outras operações.
- Medir distâncias e ângulos, criar histogramas de densidade e gráficos de perfil de linha.
- Calcular estatísticas de área e valor de pixel de seleções definidas pelo usuário.
- Realizar transformações geométricas como dimensionamento, rotação e inversão.
- Suportar qualquer número de janelas (imagens) simultaneamente, limitado apenas pela memória disponível.

O ImageJ é muito utilizado no campo das ciências biológicas, pois, além de ser um software livre, ele possui uma interface de fácil acesso e de usabilidade simples devido a abundância de documentação, tornando fácil novos usuários escreverem seus próprios *scripts*. O fato dele ser código aberto permite que desenvolvedores possam criar plugins que aumentam o repertório das funcionalidades do programa e estende a sua vida útil, contribuindo para sua popularização nos laboratórios de análise de imagem (RUEDEN et al., 2017)

2.4.3 OpenCV

Open Source Computer Vision Library ou OpenCV, é uma biblioteca de funções de programação focada em visão computacional em tempo real de código aberto em C/C++. Originalmente desenvolvido por um pequeno grupo de pesquisadores da Intel por meados de 1999, foi lançado oficialmente em 2000. Ele sofreu diversas atualizações ao decorrer do tempo, permitindo o uso de outras linguagens como Python por exemplo, e aumentando a compatibilidade com outros sistemas operacionais. (PULLI et al., 2012)

Esta biblioteca possui uma extensa lista de funções e algoritmos, ultrapassando o número de mais de 2500 algoritmos, que realizam por exemplo, reconhecimento facial, realidade virtual e até interação humano-computador. Tendo isso em mente, o OpenCV possui também uma grande variedade de módulos que auxiliam na aplicação de técnicas de PDI. (OPENCV, 2021)

2.4.4 C++

C++ é uma linguagem de programação compilada multi-paradigma e de uso geral desenvolvida por Bjarne Stroustrup em 1983 como um adicional à linguagem C. Ela é uma das

linguagens comerciais mais populares tanto comercialmente quanto academicamente, visto que ela possui um grande desempenho e grande número de ferramentas e bibliotecas disponíveis. Uma dessas ferramentas é o próprio OpenCV que foi escrita nessa linguagem.

Essa linguagem foi projetada com a ideia de orientação para programação de sistemas e softwares embarcados, que combinasse desempenho, flexibilidade de uso e eficiência. C++ conseguiu provar seu valor com seus diversos softwares e aplicativos que foram desenvolvidos nesta linguagem, como jogos, aplicações desktop e servers. (STROUSTRUP, 1997)

2.4.5 Python

A linguagem Python foi concebida no final dos anos 80, onde sua implementação começou em dezembro de 1989 por Guido van Rossum no *Centrum Wiskunde & Informatica* (Instituto Nacional de Pesquisa em Matemática e Ciência da Computação). (VENNERS, 2003) Ela por sua vez, teve sua fundação com o objetivo de substituir a linguagem de programação ABC, também criada pela CWI, onde seu foco era possuir um código mais limpo e legível, fosse de propósito geral e que pudesse dar suporte a múltiplos paradigmas de computação, como por exemplo orientação de objetos. (KUHLMAN, 2012)

Assim como C, Python possui também uma grande quantidade de bibliotecas e módulos, onde esta dispõe de um *binding* com o OpenCV, permitindo assim o uso dessa biblioteca.

2.4.6 Visual Studio Code

Visual Studio Code é um editor de texto desenvolvido e distribuído pela Microsoft de maneira gratuita para Windows, Linux e MacOS, que permite criar códigos em diversas linguagens como por exemplo C++ e Python. Dentre as diversas funcionalidades que ele disponibiliza para o programador, está a visualização da estrutura do código com identificação de erros, fluxo de dados, controle de versionamento e capacidade de depuração. (MICROSOFT, 2022)

2.4.7 Qt e Pyside

Qt ou pronunciado "*cute*" é um framework livre criado pela empresa norueguesa Trolltech que tem como objetivo auxiliar no desenvolvimento de interfaces já que com ele, é possível desenvolver aplicativos e bibliotecas uma única vez e compilá-los para diversas plataformas sem que seja necessário alterar o código fonte. (About Us, 2022)

Já o Pyside é um wrapper para a biblioteca Qt na linguagem Python, sendo ele uma alternativa de desenvolvimento de interfaces comparado ao Tkinter por exemplo, e assim como o Qt ele também é livre. (The Qt Company, 2022)

3

Metodologia

Este capítulo irá descrever e demonstrar a metodologia que foi utilizada durante o desenvolvimento do projeto através de etapas, onde para um melhor entendimento do leitor, a figura 15 exemplifica de maneira concisa as fases que foram seguidas, com uma breve descrição de cada uma delas.

Estudo das ferramentas e conceitos de PDI: Nesta fase do projeto a intenção é de ter uma maior compreensão do funcionamento das técnicas para aplicação nas imagens de tecido adiposo, na qual é esperado extrair informações da imagem. Através do estudo das ferramentas, foi possível definir a melhor abordagem para a resolução do problema.

Assim sendo, foi optado por utilizar a linguagem Python na versão 3.10.5 para o desenvolvimento deste protótipo, já que devido a sua integração com bibliotecas de PDI e módulos para gerar uma *Graphical User Interface*(GUI) de fácil codificação, a tornando altamente desejável para a criação deste sistema de forma ágil sem comprometer a usabilidade.

A escrita do código base foi feita com o auxílio do editor de texto Visual Studio Code versão 1.69.0, este que por sua vez, oferta diversas extensões e opções que contribuíram para a criação, depuração e execução do protótipo.

Dentre as bibliotecas utilizadas, cito OpenCv como sendo a mais usada no quesito de PDI, já que ela possui uma robusta gama de operações de imagens, entretanto, para a prototipação também houve a utilização de outras como a scikit, que possui módulos para o processamento de imagens, matplotlib para a visualização de dados pós processamento e numpy para operações.

Para a interface gráfica, optou-se pela utilização do QtPy, que é uma camada de abstração que permite criar aplicações utilizando apenas uma API. Com ela foi possível criar uma aplicação que atendesse os requisitos tanto para a contagem manual, quanto para a automática

Estudo dos Conceitos Biológicos: Em execução paralela a primeira fase, esta teve por objetivo adquirir um conhecimento maior sobre os conceitos biológicos relativos aos estudos de WATs e BATs como as doenças relacionadas a estes. Com isso em mente, foi possível trilhar o melhor caminho para o desenvolvimento uma solução eficiente para o problema. A professora Sabrina e, a agora mestranda, Beatriz Machado, forneceram documentos muito pertinentes sobre o processo de geração e coleta de dados referentes aos adipócitos, com o intuito de auxiliar na execução da primeira fase da construção do protótipo, de forma que fosse possível determinar qual das ferramentas pesquisadas ofertam o melhor resultado.

Implementação de um protótipo: Nesta parte do projeto, após a execução das outras duas partes, teve como por objetivo criar e desenvolver um protótipo através do uso do conhecimento adquirido, que consiga satisfazer os parâmetros exigidos para a solução do problema.

Para isso, foi criado um repositório na plataforma GitHub, com o intuito de hospedar o projeto para que pudesse desenvolvê-lo em computadores diferentes e possibilitando um download de fácil acesso para qualquer pessoa interessada em utilizá-lo.

O protótipo foi feito em dois computadores, sendo um dentro da universidade e o outro pessoal; O primeiro dispõe de um processador Intel Core i3-9100 de 3.60GHz, 8GB de memória RAM, enquanto o pessoal possui um AMD Ryzen 7 2700 de 3.6GHz (*Overclocked*) com 16GB de RAM, ambos utilizando o sistema operacional Microsoft Windows 10 de 64 bits na versão 20H2.

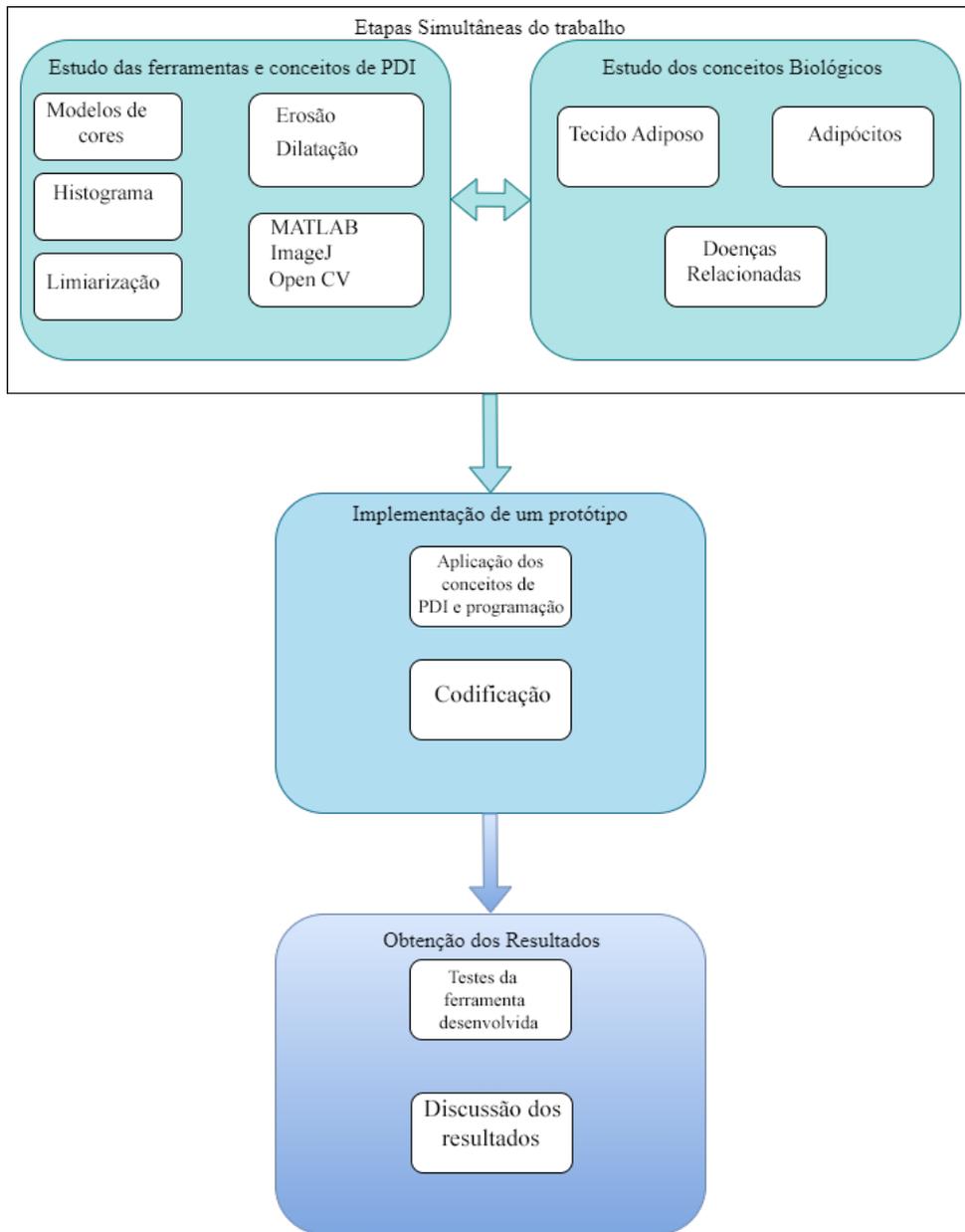
Esta parte foi descrita com maiores detalhes durante o Capítulo 4, de forma a explicar como se chegou na solução final.

Obtenção dos Resultados: E por último, foi testado o protótipo com as amostras obtidas e providenciadas pela CCBS de forma que possibilitou reunir resultados para a análise e discussão dos resultados obtidos com este, na qual será demonstrado todo o processo percorrido para chegar no objetivo final, tal que este, tem um capítulo dedicado a descrever-lo.

Estas amostras foram separadas e contabilizadas em um espelho, providenciados pela mestranda Beatriz, através do software ImageJ, onde nele é possível verificar a quantidade de adipócitos contabilizados. Por sua vez, esses resultados são utilizados como comparativo, de forma que a comparação com os dados obtidos com o processamento da imagem dentro protótipo, possa ser utilizado para uma validação.

Assim como a parte anterior, este também é abordado no Capítulo 4 com mais detalhes.

Figura 15 – Diagrama das Etapas do TCC



Fonte: Autor

4

Desenvolvimento e Resultados

Este capítulo tem por objetivo demonstrar todo o processo de desenvolvimento e pesquisa por de trás da solução final, exemplificando os passos seguidos, como também demonstrando os resultados obtidos deste, fazendo uma discussão que permite um entendimento maior sobre este trabalho.

4.1 Início do desenvolvimento

No início do desenvolvimento, seguindo as etapas da metodologia criada, foram estudados os conceitos Biológicos que envolvem os WAT como também houve um estudo das ferramentas e conceitos de PDI.

No quesito WAT, foi fornecido diversos documentos sobre o assunto, para que pudesse se obter uma maior familiariedade com os processos de criação de um rato até a análise histológica do rato. Foi verificado os laboratórios e todos os procedimentos para a geração das lâminas que são processadas posteriormente em um microscópio Olympus BX60 com uma câmera Olympus DP71 acoplada a lente, onde ela fotografa e gera as imagens com resolução de 2040x1536 pixels.

De forma simultânea, ocorrera um estudo sobre as ferramentas e conceitos de PDI, de forma que a escolha do ambiente de desenvolvimento visava atender o requisito de ser *open-source*. Apesar do MATLAB ser uma opção que poderia solucionar o problema, ofertando diversas toolboxes que permitem a criação de uma solução, esta é paga, e devido a maior familiariedade com o OpenCV comparada ao desenvolvimento de plugins para o ImageJ, o projeto foi estruturado para que fosse desenvolvido utilizando as bibliotecas do OpenCV com as operações de chamada destas sendo realizadas na linguagem Python.

4.2 Implementação de um protótipo

Tendo isso em mente, o primeiro protótipo que foi criado, utilizava um tratamento da imagem para escala de cinza, onde depois era aplicada as operações de erosão e dilatação, se faziam operações de subtração entre elas e por ultimo as binarizavam, entretanto, este se mostrava ineficaz devido ao ruído gerado por essas transformações, pois mesmo ao passar filtros para tentar mitigar o ruído destas, as bordas dos adipócitos não eram totalmente evidentes na imagem, assim comprometendo e impossibilitando a contagem como fica evidente na imagem 16.

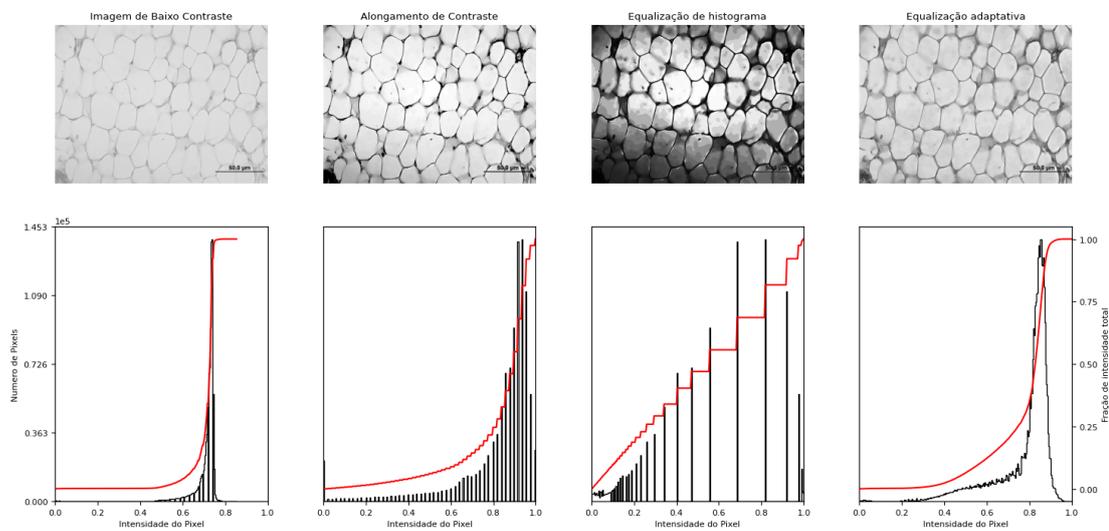
Figura 16 – Imagem demonstrando desenvolvimento inicial com imagens ruidosas



Fonte: Autor

Para tratar essas imagens ruidosas, foi construído um código que, aplicado filtros e operações de redução, seria feito a análise do histograma delas, onde aplicando diversos filtros e observando o gráfico, seria possível identificar qual das imagens exibiria um melhor resultado e poderiam ser utilizadas nas próximas fases de processamento da imagem, como é exibido na imagem 17.

Figura 17 – Imagem demonstrando histograma desenvolvido para a análise da imagem



Fonte: Autor

Dentre estas, a que se mostrou mais promissora foi a de alongamento de contraste, ou em inglês *contrast stretching*, que é uma técnica que visa melhorar o contraste esticando os valores de intensidade de uma imagem para preencher toda a gama dinâmica.

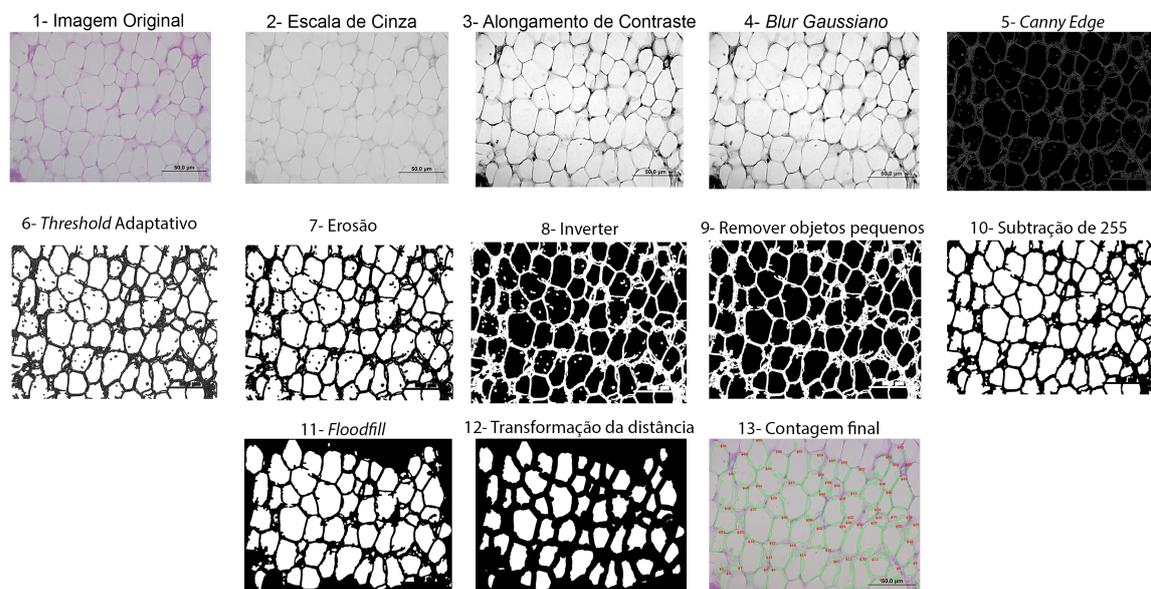
Com isso foi possível chegar no algoritmo final que consiste em:

1. Ler a imagem a ser processada
2. Transformar-la em escala de cinza para os processamentos seguintes.
3. Empregar o alongamento de contraste na imagem com o intuito de melhorar a distribuição dos valores dentro da imagem.
4. Utilizar um Blur Gaussiano para apagar objetos menores nas bordas.
5. Usar a operação de detecção de bordas de Canny buscando ressaltar os limites dos adipócitos
6. Binarizar a imagem utilizando um *Threshold* adaptativo Gaussiano.
7. Aplicar a operação de erosão na imagem para retirar partes brancas das bordas e aumentar a definição dos elementos.
8. Criar uma variável que inverta os valores binários da matriz da imagem para deixar os objetos em preto com as bordas em branco.

9. Empregar o uso de um filtro de criação própria que remove pequenos objetos, para que o adipócito fique limpo por dentro, utilizando a variável que inverteu os valores.
10. Subtrair o valor de 255 de uma operação de close da variavel invertida para que retorne os adipócitos para a cor branca e as bordas para a cor preta.
11. Criar uma borda de 1 pixel maior, onde aplicado o floodfill, irá preencher todas as células que estão adjacentes a borda da imagem para excluir-las da contagem.
12. Aplicar a transformação da distância com a binarização desta, possibilitando que adipócitos antes unidos, agora estarão separados.
13. E por último é feita a contagem e contorno dos adipócitos e salva a imagem processada.

Estes passos podem ser observados na imagem 18 mostrando iterativamente o processo até se chegar na imagem final, e também podem ser conferidas em sua integralidade no anexo 1 onde o *pipeline* do algoritmo está disponível.

Figura 18 – Passos do algoritmo sendo executados



Fonte: Autor

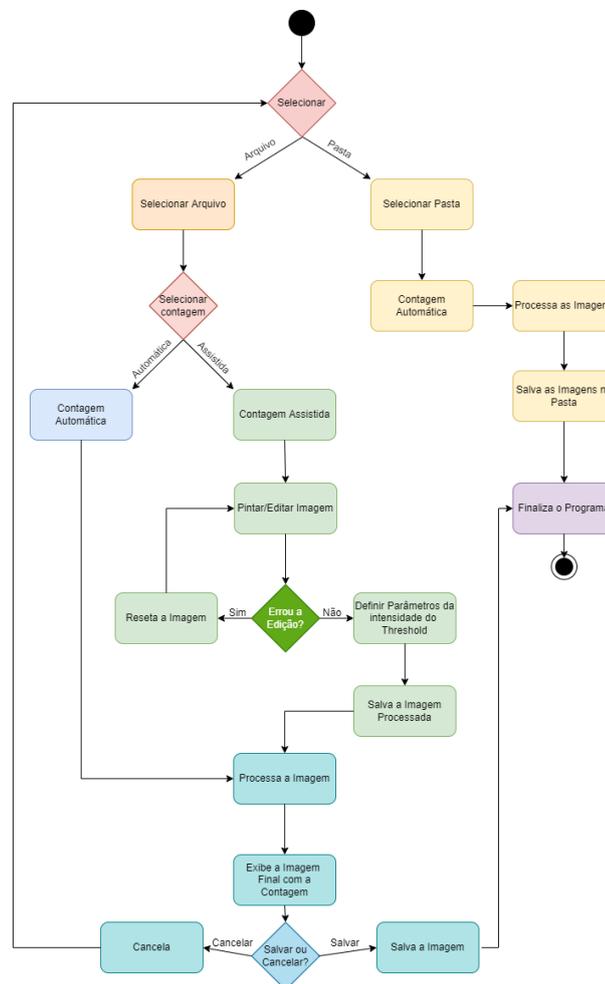
4.3 Prototipagem de uma interface

Dada a circunstância de desenvolvimento deste protótipo, se pensou na criação de uma interface, com que a pessoa que há de realizar a análise histológica do, ou seja, de forma com que

o usuário, possa interagir com o algoritmo de forma mais intuitiva comparada a uma execução via terminal.

Logo, para tal foi imaginado um diagrama base que representa o fluxo a ser seguido pelo usuário ao utilizar o programa, de forma que se leve em consideração três formas de uso, podendo ser, a leitura e contagem automática de uma imagem, a leitura e contagem automática de imagens em uma pasta, e por último, uma contagem assistida que conta com o preenchimento de áreas da imagem simulando um pincel, com a cor preta para que WAT que anteriormente não puderam ser fechados, agora possam, e com a cor branca para restaurar adipócitos muito pequenos que se perderam na contagem, tudo isso unido com o parâmetro da proporção do *threshold* da transformação da distância. A imagem 19 demonstra como esse diagrama foi pensado.

Figura 19 – Diagrama do fluxo de atividades



Fonte: Autor

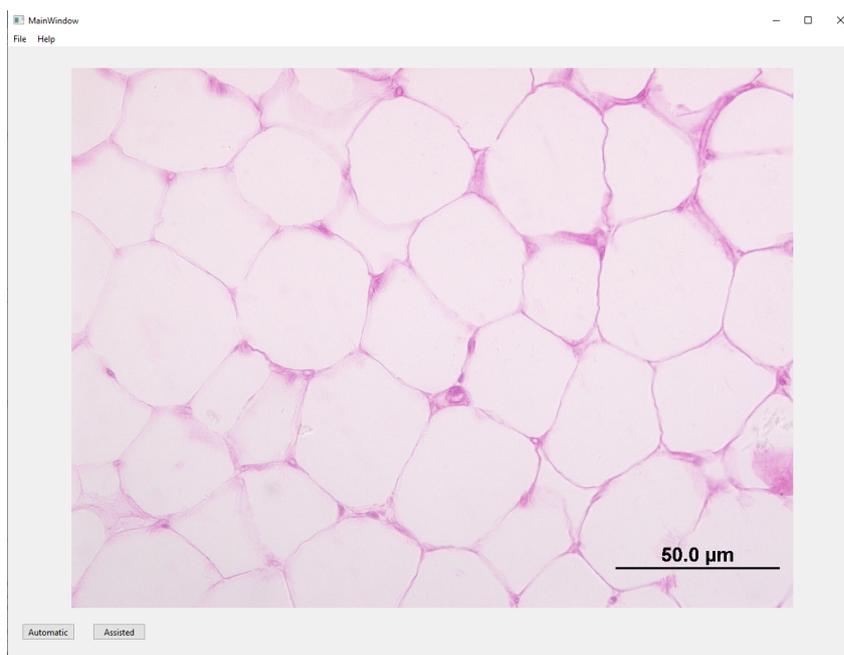
O protótipo foi apelidado de SCAPy, sendo o nome uma sigla para *Software* para Contagem de Adipócitos em Python, onde também serve para sua contraparte em inglês (*Software for Counting Adipocytes in Python*) e para o desenvolvimento da interface, foi utilizado o PySide6

que é um módulo Python oficial do projeto Qt para Python, fornecendo acesso ao framework Qt 6.0+ completo. Ele dispõe das seguintes opções:

- File: opção na barra superior que permite selecionar e ler arquivos de imagem para que esta seja carregada para dentro do programa.
- Read New File: permite que o usuário carregue uma imagem do seu computador.
- Read New Folder: permite que o usuário carregue uma pasta contendo imagens do seu computador.
- Automatic: sendo carregado uma imagem, é um botão que realiza as operações descritas anteriormente para a contagem, exibindo uma nova imagem com os dados extraídos dela.
- Assisted: é um botão que permite o processamento assistido, podendo ser alterado os parametros de *threshold* da transformação da distância e preenchimento de áreas de adipócitos que não se fecharam. Ao carregar uma pasta, esse botão fica desativado, visto que haveria um grande gargalo caso o usuário decidisse processar muitas imagens de forma assistida, já que iria consumir muito mais tempo do que simplesmente ler cada arquivo individualmente.
- Help: nele é possível encontrar algumas informações sobre o programa e funcionalidades.

Com isso em mente, estas opções podem ser visualizadas na imagem 20 para um maior entendimento.

Figura 20 – Protótipo da Interface

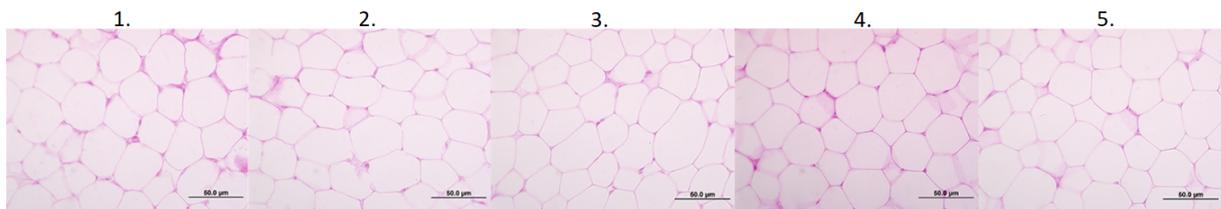


Fonte: Autor

4.4 Validação dos Resultados

A validação do protótipo foi realizada através da comparação entre um espelho fornecido pela Beatriz Daudt, na qual a contagem foi realizada utilizando o software ImageJ, e o algoritmo desenvolvido para a realização deste trabalho (SCAPy). Abaixo na imagem 21 é exibida as imagens que foram selecionadas para o teste da ferramenta e na tabela 1 é possível visualizar os resultados obtidos com o protótipo criada.

Figura 21 – Imagens Selecionadas para a validação



Fonte: Autor

Tabela 1 – Comparações de Resultados entre as Imagens

Imagem	ImageJ	SCAPy Automático	SCAPy Assistido
1	28	28	28
2	27	26	27
3	33	32	33
4	30	32	31
5	32	28	32

Apesar da contagem ser subjetiva ao analista histológico, foi utilizado como base a contagem realizada pela pessoa, e não contado novamente, sendo assim, como é possível verificar através da tabela, o software conseguiu contabilizar os adipócitos dentro da imagem com uma taxa de erro de 5% e de acerto de 95% das imagens testadas de forma automática. Enquanto utilizando a versão assistida, a taxa de acerto subiu para 98%.

5

Conclusão e Trabalhos Futuros

A ferramenta desenvolvida se demonstrou capaz de cumprir um dos objetivos de contar o número de adipócitos, dada a imagem de forma satisfatória. Apesar dos grandes empecilhos encontrados para gerar esta solução, ainda foi possível criar uma viabilidade para a resolução deste, em que, devida a vasta fonte de informações que a internet provê, pode-se afirmar que a mesma é de grande ajuda para busca de ferramentas e técnicas. Como a contagem de adipócitos com o auxílio do computador é mais específico comparado a contagem de células normais, a maioria das informações se encontram esparsas, dificultando, de certo modo, o desenvolvimento.

Entretanto, há espaço para melhoras no sistema, visto que existem diversos tipos de adipócitos, como os BAT que tendem a ter uma representação diferente no tratamento da imagem, como também outros corantes que possam modificar a imagem gerada pelo microscópio, já que isto causa uma diferença tanto na cor da imagem gerada, como no tratamento da escala de cinza e ruídos dentro dela.

Vale também resaltar que, ao comparar a imagem utilizada no SCAPy ao ImageJ, a mesma demonstra-se mais clara no primeiro programa, já que a tratada pelo outro sistema, escurece o brilho da imagem e aumenta o contraste. Isto por sua vez poderia aumentar a acurácia ao realizar os tratamentos da imagem, da mesma forma que retirar a escala de cima da imagem também ocasionaria em um aumento da precisão, visto que por esta fazer parte da imagem, causa variações ao realizar a contagem da área, e então depois de processá-la, adicionar a escala novamente.

Devido a grande quantidade de pesquisas realizadas no CCBS, é notável que existam outros tecidos sendo analisados, com padrões completamente diferentes entre eles. Então como proposta de trabalhos futuros tem-se:

1. Criação de um sistema único que compilasse diversos tipos de operações para análise histológica, visando uma maior integração entre o CCBS e o curso de Ciência da

Computação, onde iterativamente, seriam adicionadas novas "*features*" para esse sistema, sejam através de trabalhos ou iniciações científicas.

2. Adição de uma novas funcionalidade no SCAPy que permitisse calcular a área que os adipócitos ocupam dentro da imagem, já que as técnicas utilizadas para a contagem ofertaria uma solução com uma precisão baixa. Outra funcionalidade que também poderia ser adicionada, seria a contagem de BAT, já que neste também, as técnicas aplicadas para o WAT funcionariam de maneira diferente do esperado.
3. Tradução da interface para um ambiente web, onde ao invés de rodar dentro do computador do usuário, este seria um serviço que poderia ser utilizado em qualquer lugar desde que houvesse conexão com internet. Pelo fato do software ser livre, seria possível reunir todos as informações processadas dentro de um banco de dados, podendo gerar muito mais conjuntos e fazer análise geral dos mesmos.

Referências

- Robert Fisher and Simon Perkins and Ashley Walker and Erik Wolfart. *Distance Transform*. 2003. Disponível em: <<https://homepages.inf.ed.ac.uk/rbf/HIPR2/distance.htm>>. Acesso em: 13 July 2022. Citado na página 32.
- About Us. *Qt for python*. 2022. Disponível em: <<https://www.qt.io/company>>. Acesso em: 17 July 2022. Citado na página 36.
- ABRAMOWITZ, M. W. D. M. "Introduction to Microscopy". *Molecular Expressions*, National High Magnetic Field Laboratory, 1800 East Paul Dirac Dr., The Florida State University, Tallahassee, Florida, 32310., 2007. Disponível em: <<https://micro.magnet.fsu.edu/primer/anatomy/introduction.html>>. Citado na página 14.
- BORGEFORS, G. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, v. 34, n. 3, p. 344–371, 1986. ISSN 0734-189X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0734189X86800470>>. Citado na página 31.
- CANNY, J. A computational approach to edge detection. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 1986. Citado na página 32.
- CARDIFF, R.; MILLER, C.; MUNN, R. Manual hematoxylin and eosin staining of mouse tissue sections. *Cold Spring Harbor Protocols*, Cold Spring Harbor Laboratory Press, v. 2014, n. 6, p. 655–658, Jan. 2014. ISSN 1559-6095. Citado na página 20.
- CINTI, S. The adipose organ: morphological perspectives of adipose tissues. *Proceedings of the Nutrition Society*, Cambridge University Press, v. 60, n. 3, p. 319–328, 2001. Citado na página 18.
- EDINBURGH, T. U. of. "What is microscopy?". 2018. Disponível em: <<https://www.ed.ac.uk/clinical-sciences/edinburgh-imaging/for-patients-study-participants/tell-me-more-about-my-scan/what-is-microscopy>>. Citado na página 14.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 2. ed. [S.l.]: Addison-Wesley Pub, 2002. Citado na página 21.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 4. ed. 330 Hudson Street, New York, NY 10013: Pearson, 2018. Citado 11 vezes nas páginas 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 e 31.
- GRAAFF, V. de. *Anatomia Humana*. 6. ed. [S.l.]: Manole, 2003. Citado na página 18.
- HERNÁNDEZ, A. Adipose Tissue. *Elsevier, Osmosis*, 2021. Disponível em: <<https://www.osmosis.org/answers/adipose-tissue>>. Citado na página 18.
- JUNQUEIRA, L. C. U.; CARNEIRO, J. *Histologia Básica*. 9. ed. Rio de Janeiro: Guanabara Koogan, 1999. Citado na página 17.

- KLOCK, J. C. *Montagem de cariótipo de peixes assistida por computador*. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, Curitiba, mar. 2017. Citado na página 14.
- KUDA, O.; ROSSMEISL, M.; KOPECKY, J. Omega-3 fatty acids and adipose tissue biology. *Molecular Aspects of Medicine*, v. 64, p. 147–160, 2018. ISSN 0098-2997. Dietary fatty acids, lipid mediators, cell function and human health. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0098299717301620>>. Citado na página 19.
- KUHLMAN, D. *A Python Book*. [s.n.], 2012. Disponível em: <https://web.archive.org/web/20120623165941/http://cutter.rexx.com/~dkuhlman/python_book_01.html#interactive-python>. Acesso em: 11 mar 2022. Citado na página 36.
- LEESON, C. R.; LEESON, T. S.; PAPARO, A. A. *Textbook of histology*. 5. ed. Philadelphia: Philadelphia : Saunders, 1985. Disponível em: <<https://archive.org/details/textbookofhistol00lees/page/n13/mode/2up>>. Acesso em: 02 mar 2022. Citado na página 14.
- LINDSTRÖM, P. The physiology of obese-hyperglycemic mice [ob/ob mice]. *TheScientificWorldJournal*, v. 7, p. 666–85, 02 2007. Citado na página 19.
- LUBACZEUSKI, C. et al. Vagotomy ameliorates islet morphofunction and body metabolic homeostasis in msg-obese rats. *Brazilian Journal of Medical and Biological Research [online]*, v. 48, p. 447–457, 02 2015. Disponível em: <<https://www.scielo.br/j/bjmr/a/Cn4WYPChh3CcMbBYhQSZRNb/?lang=en>>. Citado na página 20.
- MATHWORKS. *MATLAB*. [S.l.], 2022. Disponível em: <<https://www.mathworks.com/products/matlab.html>>. Acesso em: 11 mar. 2022. Citado na página 34.
- MAURICIO, C. R. M. *Contador de células vermelhas baseado em imagens para múltiplas espécies de animais silvestres e domésticos*. Tese (Tese de Doutorado) — Universidade Tecnológica Federal do Paraná, Curitiba, mar. 2017. Citado na página 14.
- MCCANN, M. et al. Automated histology analysis: Opportunities for signal processing. *Signal Processing Magazine, IEEE*, v. 32, p. 78–87, 01 2015. Citado na página 14.
- MICROSOFT. *Visual Studio Code docs*. [S.l.], 2022. Disponível em: <<https://code.visualstudio.com/docs>>. Citado na página 36.
- Oak Ridge National Laboratory. *Human Genome wall for SC99*. 1999. Disponível em: <<https://www.csm.ornl.gov/SC99/GENwall.html>>. Acesso em: 12 mar 2022. Citado na página 20.
- OPENCV. *OpenCV*. [S.l.], 2021. Disponível em: <<https://opencv.org/about/>>. Citado na página 35.
- PEDRINI, H.; SCHWARTZ, W. R. *[A]nálise de imagens digitais: Princípios, Algoritmos e Aplicações*. São Paulo: Thomson Learning Edições, 2008. Citado na página 29.
- PULLI, K. A. et al. Realtime Computer Vision with OpenCV. *ACM Queue*, v. 10, 4 2012. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5708080/>>. Citado na página 35.
- RASBAND, W. *Introduction*. [S.l.], 2022. Disponível em: <<https://imagej.nih.gov/ij/docs/intro.html>>. Acesso em: 11 mar. 2022. Citado na página 35.

- REYES-FARIAS, M. et al. White adipose tissue dysfunction in obesity and aging. *Biochemical Pharmacology*, v. 192, p. 114723, 2021. ISSN 0006-2952. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0006295221003397>>. Citado 2 vezes nas páginas 17 e 19.
- RUEDEN, C. T. et al. Imagej2: Imagej for the next generation of scientific image data. *BMC Bioinformatics*, Laboratory for Optical and Computational Instrumentation, University of Wisconsin at Madison, Madison, Wisconsin USA and Morgridge Institute for Research, Madison, Wisconsin USA, v. 18, 1 2017. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5708080/>>. Citado na página 35.
- SIGGRAPH '79: Proceedings of the 6th Annual Conference on Computer Graphics and Interactive Techniques. New York, NY, USA: Association for Computing Machinery, 1979. ISBN 0897910044. Citado na página 34.
- STROUSTRUP, B. *The C++ Programming Language*. 3. ed. Massachusetts: Addison-Wesley, 1997. Disponível em: <https://archive.org/details/cprogramminglang00stro_0/page/n9/mode/2up>. Acesso em: 11 mar 2022. Citado na página 36.
- The Qt Company. *Qt for python*. 2022. Disponível em: <https://wiki.qt.io/Qt_for_Python>. Acesso em: 17 July 2022. Citado na página 36.
- VENNERS, B. The Making of Python. 01 2003. Citado na página 36.
- WANG, Q. et al. Tracking adipogenesis during white adipose tissue development, expansion and regeneration. *Nature medicine*, v. 19, 09 2013. Citado na página 19.
- World Health Organization. Obesity and overweight Fact sheet. 2015. Disponível em: <<https://www.who.int/en/news-room/fact-sheets/detail/obesity-and-overweight>>. Acesso em: 13 mar 2022. Citado na página 19.

Anexos

ANEXO A – Código dos passos do Algoritmo desenvolvido

Código 1 – Código *processing definitions*

```

1  #!/usr/bin env python3
2
3  import cv2 as cv
4  from cv2 import imwrite, imread
5  import numpy as np
6  from PySide6.QtGui import QAction, QPixmap
7  from skimage import exposure
8
9  def gray_scale_transformation(image):
10     gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
11     return gray
12
13  def contrast_stretching(gray):
14     p2, p98 = np.percentile(gray, (2, 98))
15     img_rescale = exposure.rescale_intensity(gray, in_range=(p2, p98))
16     return img_rescale
17
18  def blur(img_rescale):
19     blur = cv.GaussianBlur(img_rescale, (11,11), 0)
20     return blur
21
22  def canny_edge(blur):
23     canny = cv.Canny(blur, 0, 50)
24     return canny
25
26  def gaussian_threshold(canny):
27     thresh = cv.adaptiveThreshold(canny,255,cv.ADAPTIVE_THRESH_GAUSSIAN_C
28     ,\
29     cv.THRESH_BINARY,11,2)
30
31     return thresh
32
33  def erode(thresh):
34     kernel = cv.getStructuringElement(cv.MORPH_ELLIPSE, (3,3))
35     erosion = cv.erode(thresh, kernel,iterations = 1)
36     return erosion
37  def remove_small_objects(erosion):

```

```
38     invert = cv.bitwise_not(erosion)
39     # Filter using contour area and remove small noise
40     cnts = cv.findContours(invert, cv.RETR_TREE, cv.CHAIN_APPROX_SIMPLE)
41     cnts = cnts[0] if len(cnts) == 2 else cnts[1]
42     for c in cnts:
43         area = cv.contourArea(c)
44         if area < 5500:
45             cv.drawContours(invert, [c], -1, (0,0,0), -1)
46     return invert
47
48 def closing(invert):
49     # Morph close and invert image
50     kernel = cv.getStructuringElement(cv.MORPH_RECT, (5,5))
51     close = 255 - cv.morphologyEx(invert, cv.MORPH_CLOSE, kernel,
52     iterations=2)
53     return close
54
55 def flood_fill(close, image):
56     pad = cv.copyMakeBorder(close, 1,1,1,1, cv.BORDER_CONSTANT, value=255)
57     h, w = pad.shape
58
59     mascara = np.zeros([h + 2, w + 2], np.uint8)
60
61     # floodfill outer white border with black
62     img_floodfill = cv.floodFill(pad, mascara, (0,0), 0, (5), (0), flags=8)
63     [1]
64
65     # remove border
66     img_floodfill = img_floodfill[1:h-1, 1:w-1]
67
68     # save cropped image
69     cv.imwrite('floodfilled.jpg',img_floodfill)
70
71     (cnt, hierarchy) = cv.findContours(
72     img_floodfill.copy(), cv.RETR_EXTERNAL, cv.CHAIN_APPROX_NONE)
73     rgb = cv.cvtColor(image, cv.COLOR_BGR2RGB)
74     cv.drawContours(rgb, cnt, -1, (0, 255, 0), 2)
75
76     cv.imwrite("contorno_celula.jpg", rgb)
77     return img_floodfill, rgb
78
79 def dist_transform_plus_thresh(thresh_proportion: int = 0.130 ,write: bool
80 = False):
81     floodfilled = cv.imread("floodfilled.jpg")
82
83     ff_gray = cv.cvtColor(floodfilled, cv.COLOR_BGR2GRAY)
```

```
82     ff_dist = cv.distanceTransform(ff_gray, cv.DIST_L2, 0)
83     if write: cv.imwrite("ff_dist.png", ff_dist)
84
85     ret, ff_bin = cv.threshold(ff_dist, thresh_proportion*ff_dist.max(),
86                               255, 0)
87     if write: cv.imwrite("ff_bin.png", ff_bin)
88     return ff_dist, ff_bin
89 def count_it(ff_dist, ff_bin, processed_image):
90
91     ff_dist, ff_bin = dist_transform_plus_thresh(0.130, False)
92
93     image = cv.imread("floodfilled.jpg")
94
95     (cnt, hierarchy) = cv.findContours(
96         ff_bin.copy().astype(np.uint8), cv.RETR_EXTERNAL, cv.
97         CHAIN_APPROX_NONE)
98     rgb = cv.cvtColor(image, cv.COLOR_BGR2RGB)
99     cv.drawContours(rgb, cnt, -1, (0, 255, 0), 2)
100     image_f = processed_image
101
102     c = 1
103     #cv.drawContours(rgb, cnt, -1, (0, 255, 0), 2)
104     font = cv.FONT_HERSHEY_COMPLEX
105     # Going through every contours found in the image.
106     for cnts in cnt :
107         approx = cv.approxPolyDP(cnts, 0.009 * cv.arcLength(cnts, True),
108                                 True)
109
110         # draws boundary of contours.
111         cv.drawContours(rgb, cnt, -1, (0, 255, 0), 2)
112
113         # Used to flattened the array containing
114         # the co-ordinates of the vertices.
115
116         n = approx.ravel()
117         i = 0
118
119         for j in n :
120             if(i % 2 == 0):
121                 x = n[i]
122                 y = n[i + 1]
123
124                 # String containing the co-ordinates.
125                 string = str(x) + " " + str(y)
```

```
126
127         if(i == 0):
128             # text on topmost co-ordinate.
129             cv.putText(image_f, "{}".format(c), (x, y),
130                        font, 1, (0, 0, 255), 2)
131             c = c+1
132         i = i + 1
133
134     cv.imwrite("contorno.jpg", rgb)
135     cv.imwrite("final.jpg", image_f)
136     return len(cnt)
```