

# Plataforma Semi-Automática para Segmentação de Imagens Biológicas com Auxílio de Algoritmos Bio-Inspirados

Lucas Pampolin Laheras (lucaslaheras@hotmail.com)

Gilson Antonio Giraldi (gilson@lncc.br)

Paulo Sérgio Silva Rodrigues (psergio@fei.edu.br)

Francisco José Pereira Lopes (flopes@ufrj.br)

São Bernardo do Campo, SP

Fevereiro de 2020

## 1 Introdução

Avanços significativos na geração de imagens nas áreas de medicina e biologia têm gerado uma forte demanda pelo desenvolvimento de tecnologias relacionadas à digitalização automática, tanto em hardware quanto em software. Por conta disso, o desenvolvimento das áreas de processamento de imagens, inteligência artificial, aprendizagem profunda e ciência de dados é estratégico, particularmente para a biológica, uma vez que atualmente tanto hardware quanto software têm auxiliado especialistas e agilizado os procedimentos.

O desenvolvimento e recentes avanços na microscopia confocal por varredura a laser, associado ao uso de marcadores fluorescentes, têm possibilitado a geração de dados a cerca dos níveis de expressão gênica e localização proteica tanto em células fixadas quanto in vivo. [21].

Estima-se que cerca de 75 por cento dos genes associados a doenças em humanos tenham homólogos em *Drosophila melanogaster* (mosca da fruta) [17]. Essa grande semelhanças entre os genes em humanos e em *Drosophila*, bem como o seu rápido ciclo de desenvolvimento, têm contribuído para que esse organismo seja um dos mais importantes modelos biológicos nos mais diversos campos da biologia e medicina. Dentre os estudos se destacam Alzheimer [5], Câncer [16, 20], doenças neurodegenerativas [6], vírus [7], para citar alguns. A Figura 1 demonstra as imagens utilizadas no aplicativo, são células do ovário de *Drosophilas* produzidas no Grupo de Biologia do Desenvolvimento e Sistemas Dinâmicos da Universidade Federal do Rio de Janeiro (UFRJ). A análise dessas imagens (Figura 1) tem por objetivo a segmentação das diversas células marcadas com o objetivo de identificar as células-tronco embrionárias responsável pela formação do tecido.

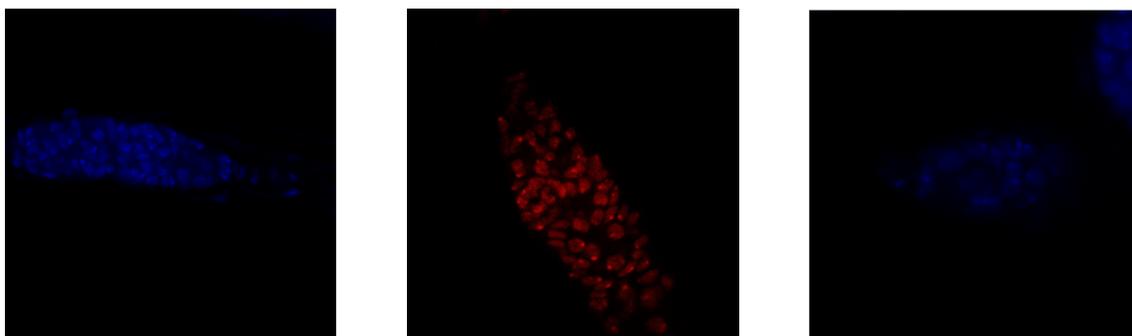


Fig. 1: Exemplos de imagens de microscopia confocal do ovário de *Drosophila*. Marcação para o núcleo das células (cor azul, marcado DAPI) e para uma proteína Adducin (vermelho, marcação de membrana).

Recentemente, a literatura de processamento digital de imagens e visão computacional tem

demonstrado a crescente efetividade da aplicação de algoritmos bio-inspirados para segmentação de imagens [13, 15, 22], bem como no método de Level Set.

Por outro lado, métodos em aprendizagem profunda, baseados em rede neural convolucional (*Convolutional Neural Network* - CNN) foram também utilizados para segmentação de imagens, obtendo resultados muito promissores [19]. Vale ressaltar que a área de aprendizagem profunda vem ganhando espaço em inúmeras aplicações em reconhecimento de padrões [3], jogos [9], veículos autônomos [11], além de outras [14]. Porém, o treinamento de uma CNN para segmentação, em geral, necessita de grandes bancos de imagens contendo os dados originais e o padrão-ouro (imagem segmentada por especialista), que na maioria das vezes não estão disponíveis.

Há diversas maneiras de aumento de dados para imagens para o uso de CNN. Entre as que mais se destacam está a criação de imagens artificiais [18] por meio das imagens originais. Por exemplo, a aplicação de zoom, rotação, flipe horizontal e vertical na imagem. Também é possível usar a estratégia conhecida como transferência de aprendizado [8] que reutiliza uma CNN treinada anteriormente sobre um grande conjunto de treinamento supervisionado obtido de outro problema semelhante.

Porém, para ambos os casos é necessário que um especialista na área tenha gerado o padrão ouro (imagem corretamente segmentada e classificada), que na maioria das vezes não estão disponíveis, uma vez que demandam muito tempo do especialistas e técnicas apuradas para análise manual de um conjunto geralmente grande de amostras.

Atualmente, não existem *pipelines* de segmentação genérico que possam ser utilizados em todos os tipos de aplicações. Mesmo em um âmbito específico, como segmentação de células, poucos *pipelines* conseguem chegar automaticamente a um resultado ideal. Por outro lado, ruídos e problemas de amostragem podem degradar os resultados obtidos através das técnicas tradicionais de segmentação de imagem. Nesse sentido, há a possibilidade de utilizar essa vasta quantidade de *pipelines* para ajudar o especialista na geração do padrão ouro por meio de uma plataforma de segmentação manual. Essa plataforma pode executar um *pipeline* escolhido que se adéqua ao tipo de imagem utilizada. Assim, uma vez executado o pipeline automaticamente, o especialista é capaz de manualmente resolver os problemas da segmentação, em um processo que aproveita o resultado parcial dos algoritmos de processamento de imagens para diminuir o trabalho final do especialista.

Assim, o objetivo principal deste trabalho é desenvolver um software que incorpore um *framework* semi-automático para a segmentação de imagens biológicas de microscopia de óvulos de *Drosophilas*. Com este aplicativo, poderemos gerar imagens anotadas (padrão-ouro) que possam ser utilizadas para treinamento de Redes Neurais artificiais, tais como a U-net e V-net [19, 12], que podem aprender a realizar a segmentação completa e automática das imagens.

## 2 Referencial

Para a implementação da plataforma foi utilizado um módulo padrão do Python para interface gráfica multiplataforma chamado TKinter [2]. Essa plataforma irá executar o pipeline desenvolvido, explicado em 3, que utiliza dois algoritmos fundamentais: o algoritmo bio-inspirado *Firefly* [23], que inicializa o algoritmo de segmentação final *Level Set* [4].

### 2.1 Firefly

O algoritmo do *Firefly*, proposto por Xin-She Yang em [23], é uma meta-heurística inspirada pelo comportamento dos vaga-lumes, que são atraídos um pelo outro de acordo com sua luminosidade natural. No final, a convergência é alcançada gerando aglomerados de vaga-lumes, onde os mais brilhantes atraem os outros vaga-lumes. Se um vaga-lume em particular é o mais brilhante, ele se moverá aleatoriamente.

A ideia geral é modelar um problema de otimização não linear associando à variável de cada problema a um vaga-lume e fazer a avaliação objetiva dependendo dessas variáveis, que estão associadas às intensidades dos brilhos dos vaga-lumes. Então, iterativamente, as variáveis são atualizadas (seus brilhos) sob regras preestabelecidas até a convergência para um mínimo global. Genericamente, em cada geração, são realizadas as seguintes etapas: avaliar o brilho, calcular todas as distâncias entre cada par de vaga-lumes, mover cada um dos vaga-lumes em direção aos outros de acordo com o brilho, manter a melhor solução (o vaga-lume mais brilhante) e gerar aleatoriamente novas soluções.

O núcleo do algoritmo é sua função de avaliação  $Z$ , que depende do problema atual, especificamente o problema de limiarização de vários níveis (*MultiLevel Thresholding Problem - MLOTP*), cada vaga-lume é considerado uma variável  $d$ -dimensional, onde para cada dimensão é computado um limiar distinto, particionando o espaço do histograma.

### 2.2 Algoritmo *Level Set*

O conceito básico deste algoritmo é delimitar a fronteira de uma região planar utilizando o conjunto de pontos no nível zero de uma função  $\alpha : \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}$  dado pela seguinte expressão:

$$S(t) = \{(x, y) \in \mathbb{R}^2 \mid \alpha(x, y, t) = 0\} \quad (1)$$

Dada uma curva de nível zero  $S(t)$  no instante  $t$ , consideram-se três possíveis valores para a função. Quando o valor é menor que 0 o ponto  $(x, y)$  está dentro da superfície  $S$ , quando o valor é maior que 0 o ponto  $(x, y)$  está fora da superfície; e quando o valor for igual a 0 o ponto

$(x, y)$  pertence a curva  $S(t)$ .

O próximo passo é encontrar a formulação euleriana da evolução da frente  $S(t)$ , gerando uma equação diferencial parcial que descreve a evolução da curva de nível no tempo e no espaço [10].

### 3 Metodologia

O presente trabalho envolve as seguintes etapas: (a) Desenvolvimento de um pipeline específico para segmentação de imagens de microscopia confocal de óvulos de *Drosophilas*; (b) Desenvolvimento de uma plataforma interativa para segmentação das imagens de interesse.

A etapa (a) está fundamentada no algoritmo Firefly (Seção 2.1), utilizando também técnicas baseadas em métodos de contorno ativo, tais como *Level Set* (Seção 2.2), para gerar um pipeline que faça pre-segmentação automática da imagem. Na etapa (b) será utilizado um modelo de interface que permita ao usuário corrigir os erros de segmentação da etapa anterior.

#### 3.1 Desenvolvimento do Pipeline

O *pipeline* exibido na Figura 2 foi implementado em Python, sendo composto por 6 módulos principais, descritos em mais detalhes a seguir.



Fig. 2: Pipeline utilizado para pre-segmentação das imagens.

O primeiro bloco (bloco 1, Figura 2) converte a imagem original, mostrada na Figura 3.(a), para tons-de-cinza. Pode-se observar o fraco contraste entre os objetos e o fundo da imagem, o que dificulta o tratamento automático destas imagens.

Na etapa correspondente ao bloco 2 da Figura 2, a imagem fornecida pela etapa 1 é processada pelo algoritmo bio-inspirado *Firefly* (Seção 2.1). A saída dessa imagem é mostrada na Figura 3.(b).

No passo correspondente à construção da máscara (bloco 3, Figura 2) a imagem de saída da etapa 2 é composta por  $N \geq 1$  regiões de interesse. Se a área de uma dessas regiões for maior que um limiar  $L > 0, L \in \mathbb{N}$ , então essa região é definida como sendo uma máscara binária. Sendo assim, a saída dessa etapa é o conjunto  $S = \{m_1, m_2, \dots, m_N\}$  de máscaras binárias. Em seguida, é aplicado três vezes a operação de dilatação elemento estruturante 3x3 em todo conjunto  $S$  gerado. Esta etapa está representada na Figura 3.(c).

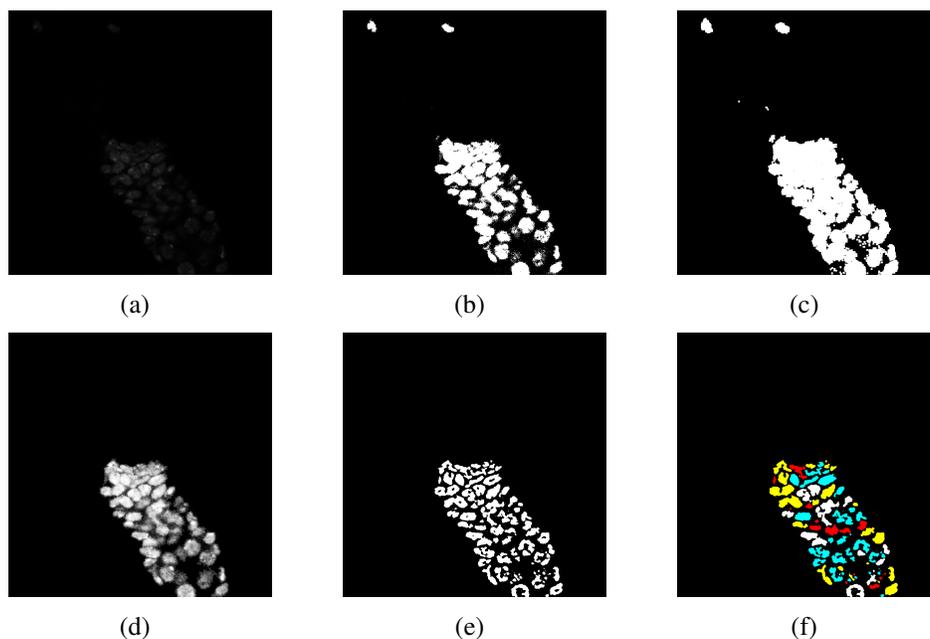


Fig. 3: (a) Imagem original. (b) Firefly aplicado na imagem após ajuste de contraste. (c) Mascaras delimitando as regiões obtidas. (d) Equalização do histograma aplicado localmente. (e) Resultado da aplicação do método de Level set 2.2. (f) Composição e coloração das regiões.

E selecionada a máscara com a maior área, a qual é utilizada para a aplicação do histograma local (Bloco 4, Figura 2), sobre a imagem original. A razão da aplicação da equalização local é o fato de que, localmente, as regiões inhomogêneas são reduzidas. A Figura 3.(d) é o resultado da equalização da Figura 3.(a) utilizando a máscara encontrada na Figura 3.(c).

Na etapa referente ao bloco 5 da Figura 2, para cada imagem de saída da etapa anterior é utilizado o algoritmo *Level Set* (Seção 2.2) de forma a delimitar as regiões locais, para obter um resultado melhor na contagem de células, como pode ser visto na Figura 3.(e).

Na última etapa (bloco 6, Figura 2), será feito o agrupamento de cada imagem de saída da etapa anterior em uma única imagem. Após o agrupamento, a cada região é atribuída uma cor aleatoriamente, de maneira que possam ser distinguidas visualmente, como pode ser observado na Figura 3.(f).

### 3.2 Desenvolvimento do *Segmentation Interface*

Foi feito o levantamento e análise de requisitos junto com biólogos da universidade Federal do Rio de Janeiro, para saber quais são as necessidades para efetuar uma segmentação adequada.

O objetivo principal deste trabalho é criar um ambiente computacional eficiente e facil-

mente expansível para o segmentação de imagens biológicas. O desenvolvimento do software foi direcionado seguindo a ideia de utilizar bibliotecas de código aberto, possibilitando um desenvolvimento rápido e com baixo custo. Assim, optamos pela biblioteca `opencv-python` [1] para processamento de imagem e `TKinter` [2] para a criação da interface.

Considerando o objetivo fundamental do sistema, podemos afirmar que o fluxo de dados no *Segmentation Interface 4* envolve basicamente leitura, processamento, visualização e modificação de imagens. As bibliotecas Python escolhidas para estas tarefas (`TKinter`, `PIL` e `opencv-python`) possuem formatos de dados próprios. Assim, um ponto fundamental no desenvolvimento do sistema é a escolha de um formato padrão para representar as imagens. Foi determinado o formato "matriz" da biblioteca `NumPy`, pois desta forma podemos utilizar a biblioteca `opencv-python` para os recursos de processamento e , após transformar a imagem para o formato `PIL`, para mostrar a imagem na saída gráfica da biblioteca `TKinter`.

## 4 Resultados

O *pipeline* de segmentação descrito acima foi implementado em uma plataforma interativa (Figura 4) para que os biólogos das equipes envolvidas possam determinar o padrão ouro das imagens que farão parte do banco de dados utilizado para o treinamento da CNN.

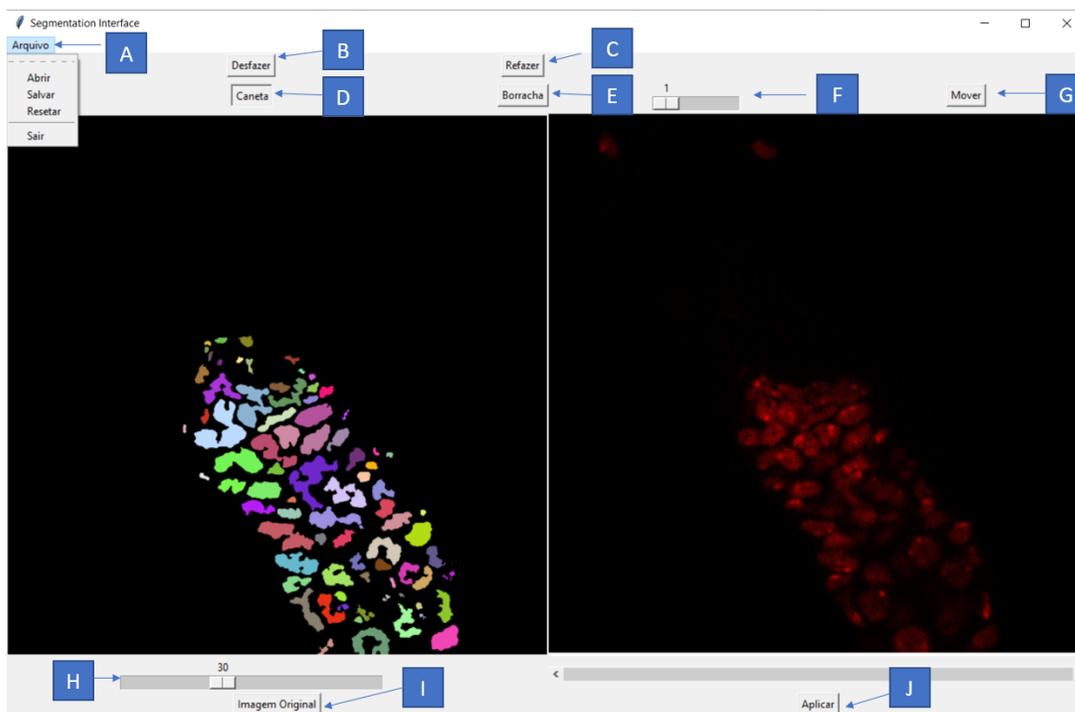


Fig. 4: Leiaute da interface gráfica desenvolvida.

Para fazer uma alteração na imagem segmentada pelo pipeline (imagem da esquerda da Figura 4) é necessário clicar com o *mouse* na área que precisa ser corrigida e o contorno dessa área será destacado na imagem da direita (imagem original dos dados na entrada), possibilitando ao usuário com o uso das ferramentas de caneta (Figura 4.D) e borracha (Figura 4.D) modificar ou remover o contorno e aplicar (Figura 4.J) na imagem segmentada, substituindo a área selecionada anteriormente pela área dentro do contorno modificado.

Para facilitar a seleção para o especialista, foi implementado o *scroll* (Figura 4.H), que permite determinar um limiar, no qual, todas as áreas menores que esse valor serão consideradas ruído e removidas da imagem segmentada. Para fornecer uma segmentação mais precisa, o usuário pode aplicar zoom na imagem da direita, modificar o tamanho da caneta e borracha com o *scroll* (Figura 4.F) e, com a ajuda do zoom aplicado, mover para a região de interesse da imagem usando o botão "Mover"(Figura 4.G). Caso ocorra algum erro na segmentação, o usuário pode desfazer ou refazer uma alteração que foi feita pelo mesmo.

O salvamento da imagem segmentada será efetuado na mesma pasta em que a imagem original foi aberta, sendo o nome padronizado com o nome da imagem original e finalizando com "PO.png", pois caso ocorra uma interrupção no processo da segmentação manual e a imagem tenha sido salva, quando a imagem original for re-aberta não será necessário o uso do *pipeline* de segmentação novamente. No entanto, caso haja necessidade, é possível forçar clicando em "Resetar"(terceira opção da Figura 4.A).

## 5 Conclusão

O *pipeline* desenvolvido obteve resultados satisfatórios até o momento, permitindo, com a integração da plataforma gráfica, a integração das imagens para que o especialista possa avaliar os resultados.

O projeto está em fase de coleta de informações por biólogos para a geração de banco de dados anotado desse tipo de imagem, que permitirá o treinamento de sistemas de aprendizado de máquina. O resultado específico apresentado na Figura 4-lado esquerdo, indica que a metodologia apresentada aqui conseguiu contrastar regiões de interesse que ocorrem na imagem da Figura 4-direita, mas que não poderiam ser visualizadas sem a metodologia proposta.

Agradecemos ao CNPq (Projeto 155130/2019-6), assim como à FEI (Fundação Educacional Inaciana), ao LNCC (Laboratório Nacional de Computação Científica) e à UFRJ (Universidade Federal do Rio de Janeiro), pelo suporte deste trabalho.

## Referências

- [1] Opencv modules.
- [2] tkinter - python interface to tcl/tk.
- [3] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.
- [4] Fangfang Dong, Zengsi Chen, and Jinwei Wang. A new level set method for inhomogeneous image segmentation. *Image and Vision Computing*, 31(10):809–822, 2013.
- [5] James P Higham, Bilal R Malik, Edgar Buhl, Jennifer M Dawson, Anna S Ogier, Katie Lunnion, and James JL Hodge. Alzheimer’s disease associated genes ankyrin and tau cause shortened lifespan and memory loss in drosophila. *Frontiers in cellular neuroscience*, 13, 2019.
- [6] Frank Hirth. Drosophila melanogaster in the study of human neurodegeneration. *CNS & Neurological Disorders-Drug Targets (Formerly Current Drug Targets-CNS & Neurological Disorders)*, 9(4):504–523, 2010.
- [7] Tamara T Hughes, Amanda L Allen, Joseph E Bardin, Megan N Christian, Kansei Daimon, Kelsey D Dozier, Caom L Hansen, Lisa M Holcomb, and Joseph Ahlander. Drosophila as a genetic model for studying pathogenic human viruses. *Virology*, 423(1):1–5, 2012.
- [8] Benjamin Q Huynh, Hui Li, and Maryellen L Giger. Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. *Journal of Medical Imaging*, 3(3):034501, 2016.
- [9] Niels Justesen, Philip Bontrager, Julian Togelius, and Sebastian Risi. Deep learning for video game playing. *IEEE Transactions on Games*, 2019.
- [10] Ravi Malladi, James A Sethian, and Baba C Vemuri. Shape modeling with front propagation: A level set approach. 1994.
- [11] Rowan McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. *International Joint Conferences on Artificial Intelligence, Inc.*, 2017.
- [12] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016.
- [13] Hongwei Mo, Lifang Xu, and Mengjiao Geng. Image segmentation based on bio-inspired optimization algorithms. pages 259–284, 01 2014.
- [14] Philosophia Naturalis. 30 amazing applications of deep learning, Jun 2018.
- [15] Valentín Osuna-Enciso. Bioinspired metaheuristics for image segmentation. 2014.

- 
- [16] O. F. Palmeira. Efeitos da irradiação em células-tronco germinativas de *Drosophila melanogaster* e influência da dieta na recuperação dessas células e tecidos ovarianos submetidos ao tratamento com cisplatina. Monografia apresentada ao Programa de Graduação em Ciências Biológicas: Biotecnologia, do Polo Avançado de Xerém, Universidade Federal do Rio de Janeiro, 2017. UFRJ.
- [17] Udai Bhan Pandey and Charles D Nichols. Human disease models in *Drosophila melanogaster* and the role of the fly in therapeutic drug discovery. *Pharmacological reviews*, 63(2):411–436, 2011.
- [18] Wilburn E Reddick, John O Glass, Edwin N Cook, T David Elkin, and Russell J Deaton. Automated segmentation and classification of multispectral magnetic resonance images of brain using artificial neural networks. *IEEE Transactions on medical imaging*, 16(6):911–918, 1997.
- [19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [20] M Sonoshita and RL Cagan. Modeling human cancers in *Drosophila*. In *Current topics in developmental biology*, volume 121, pages 287–309. Elsevier, 2017.
- [21] David J Stephens and Victoria J Allan. Light microscopy techniques for live cell imaging. *science*, 300(5616):82–86, 2003.
- [22] G. A. Wachs Lopes, F. S. Beltrame, R. M. Santos, and P. S. Rodrigues. Comparison of bio-inspired algorithms from the point of view of medical image segmentation. In *2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, pages 1–7, July 2018.
- [23] Xin-She Yang. Firefly algorithms for multimodal optimization. In *SAGA*, 2009.