

UNIVERSIDADE DO ESTADO DO AMAZONAS – UEA
ESCOLA SUPERIOR DE TECNOLOGIA – EST
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

BERNARDO DOS SANTOS FRÓES

**DESENVOLVIMENTO DE ALGORITMO COMPUTACIONAL PARA A
DETECÇÃO E A CONTAGEM DE OVOS DO MOSQUITO *ANOPHELES***

MANAUS-AM

2024

BERNARDO DOS SANTOS FRÓES

**DESENVOLVIMENTO DE ALGORITMO COMPUTACIONAL PARA A
DETECCÃO E A CONTAGEM DE OVOS DO MOSQUITO *ANOPHELES***

Trabalho de Conclusão de Curso apresentado à banca avaliadora do Curso de Engenharia de Controle e Automação da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro de Controle e Automação.

Orientador: Almir Kimura Junior

MANAUS-AM

2024

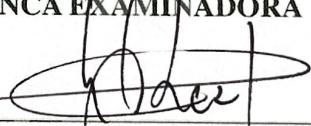
BERNARDO DOS SANTOS FRÓES

**DESENVOLVIMENTO DE ALGORITMO COMPUTACIONAL PARA A
DETECÇÃO E A CONTAGEM DE OVOS DO MOSQUITO *ANOPHELES***

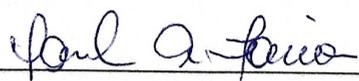
Trabalho de Conclusão de Curso
apresentado à banca avaliadora do Curso
de Engenharia de Controle e Automação
da Universidade do Estado do
Amazonas, como pré-requisito para
obtenção do título de Engenheiro de
Controle e Automação.

Aprovado em 25 de julho de 2024.

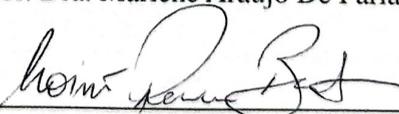
BANCA EXAMINADORA



Prof. MSc. Cleto C. de S. Leal



Prof. Dra. Marlene Araújo De Faria



Prof. MSc. Moisés Pereira Bastos

MANAUS-AM

2024

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a Deus por ter me acompanhado durante toda minha jornada acadêmica, me abençoando e me guiando pelo melhor caminho, tornando esse momento possível.

À Universidade do Estado do Amazonas – UEA, por ter sido como uma “casa” para mim durante todos esses anos e por ter sido a responsável direta por todo o conhecimento obtido para a realização desse trabalho.

À minha família, que sempre me apoiou e esteve ao meu lado nos momentos bons e nos momentos difíceis. Em especial às minhas irmãs Ana Cristina Fróes e Cristiane Fróes; à minha mãe Isis dos Santos Fróes; ao meu pai, que nos deixou em 2014, mas posso garantir que esteve presente todo esse tempo me acompanhado de outro plano, Geraldo Gonçalves Fróes.

A todos os professores que eu tive o prazer de compartilhar esses anos, tentando sempre absorver o máximo de conhecimento possível, não somente nas disciplinas, mas também em ensinamentos que guardo para a vida. Gostaria de citar dois professores que foram essenciais na minha formação: o professor Almir Kimura Júnior, que me orientou em relação ao desenvolvimento desse trabalho e ajudou na minha formação, contribuindo também para a minha preparação para o mercado de trabalho; e ao professor Charles Melo, que hoje também nos acompanha em outro plano, mas que em vida ajudou esse momento a se tornar possível.

Aos meus amigos, tanto aqueles que estiveram presentes até o final da minha jornada acadêmica, quanto aqueles que por algum motivo se distanciaram com o passar dos anos, deixo aqui meu muito obrigado por terem sido alguém que pude contar nos momentos difíceis, sobretudo àqueles que estiveram presentes nos momentos em que a depressão e a ansiedade se tornaram grandes vilões. Aqui, porém, não irei citar nomes para não ser injusto com aqueles que não lembrei, uma vez que fiz muitas amizades durante esse período na UEA, além dos amigos que cultivei fora da universidade.

À doutora Sabrina Norat Bastos, minha psicóloga, e ao meu mestre e professor de *muay thai* Rafael Guilherme, que me ajudaram a manter minha saúde mental e física em ordem sobretudo nesses últimos anos.

A todos vocês, minha eterna gratidão por estarem presentes na minha vida e por todos os momentos que pude compartilhar durante todos esses anos.

RESUMO

Atualmente, a malária é uma doença potencialmente grave, podendo levar a morte e sendo transmitida ao homem na maioria das vezes através da picada de mosquitos do gênero *Anopheles* após infectados por um parasita do gênero *Plasmodium*. O presente estudo teve como objetivo geral desenvolver um algoritmo computacional capaz de identificar e contar ovos do mosquito transmissor da doença a fim de ser utilizado em locais cuja presença de água parada possa se tornar um risco para a transmissão e proliferação da malária. O método adotado neste estudo seguiu um processo bem definido dentro do *Google Colab*, começando pela importação de bibliotecas cruciais como *cv2*, *numpy*, *keras* e *matplotlib*, necessárias para o tratamento de imagens e modelagem de aprendizado de máquina. Inicialmente, as imagens foram lidas e convertidas para escala de cinza para simplificar o processamento. Em seguida, utilizou-se a limiarização de Otsu para segmentação, melhorada com a aplicação de um filtro de desfoque gaussiano para atenuar ruídos. Uma segunda rodada de limiarização de Otsu aprimorou a segmentação pós-desfoque. Com as imagens preparadas, um modelo de rede neural pré-treinado foi empregado para identificar e classificar os padrões de interesse, enquanto um espaço de dados foi estabelecido para o armazenamento e análise das imagens processadas, configurando uma base sólida para a contagem automatizada e análise subsequente dos ovos de mosquito identificados. Os resultados do algoritmo de detecção de ovos de mosquito *Anopheles*, aplicado a um conjunto de 13 imagens, mostraram uma média de acerto de 30,35% e uma taxa de erro de 69,65%. Algumas imagens de ovos isoladas tiveram um desempenho de detecção perfeito (100% de acerto), enquanto outras, dispostas em um conjunto separado e com uma maior quantidade de objetos presentes, apresentaram desafios significativos, resultando em baixas taxas de acertos, como 6,06%. Isso indica que ainda há espaço para melhorias em relação à identificação dos falsos positivos presentes nas imagens e em relação a sua diferenciação na comparação com os ovos originais (verdadeiros positivos).

Palavras-chaves: Processamento Digital de Imagens. *Anopheles*. Limiarização de Otsu. *Machine Learning*.

ABSTRACT

Currently, malaria is a potentially serious sickness that can lead to death and being transmitted to the man mostly through *Anopheles* gender mosquito bites, after being infected by a *Plasmodium* gender parasite. The general objective of the present study was to develop a computational algorithm capable of identifying and counting the eggs of the mosquito that transmits this disease in order to be used in places where the presence of stagnant water may become a risk for the transmission and proliferation of malaria. The adopted method in this study followed a well-defined process within *Google Colab*, starting with the import of crucial libraries such as *cv2*, *numpy*, *keras* and *matplotlib* necessary for image treatment and machine learning modeling. Initially, the images were read and converted to grayscale to simplify the processing. Next, the Otsu thresholding was used for segmentation improved by applying a Gaussian filter to attenuate noises. A second round of Otsu thresholding improved the post-blurring segmentation. With the images prepared, a pre-trained neural network model was used to identify and classify patterns of interest, while a data space was established for the storage and analysis of the processed images. The results of the algorithm of the mosquito *Anopheles* egg detection, applied to a group of 13 images, showed a average hit rate of 30.35% and an error rate of 69.65%. Some isolated egg images had a perfect detection performance (100% accuracy), while others, arranged in a separated set and with a higher number of objects presented significant challenges, resulting in lower hit rates, such as 6.06%. This indicates that there is still space for improvements related to the identification of false positives present in the images and related to its differentiation comparing to the original eggs (true positives).

Keywords: Digital Image processing. *Anopheles*. Otsu thresholding. Machine Learning.

LISTA DE FIGURAS

Figura 1: Etapas fundamentais do Processamento Digital de Imagens (Gonzales & Woods, 2000).	17
Figura 2: Uma imagem monocromática e a convenção utilizada para o par de eixos (x,y). (Marques Filho & Vieira Neto, 1999).....	20
Figura 3: Modelo RGB (Pedrini & Schwartz, 2008).	20
Figura 4: Círculos cromáticos RGB (Rocha, 2010).	21
Figura 5: Representações de uma imagem: a) cores (formato RGB) e b) escalas de cinza (Peixoto, 2010).	21
Figura 6: Uma imagem 3x5 e seu respectivo histograma (Souza & Correia, 2007).....	23
Figura 7: Filtros passa-baixa de média do tipo 3x3, 5x5 e 7x7 (Oliveira et al., 2004).	25
Figura 8: Aplicação do filtro de média: (a) imagem original; (b) imagem com ruídos atenuados; (c) imagem filtrada com uma máscara 3x3 e (d) imagem filtrada com uma máscara 5x5 (Queiroz & Gomes, 2001).....	25
Figura 9: Aplicação do filtro Gaussiano: (a) imagem original e (b) imagem após a aplicação de um filtro Gaussiano (Oliveira & Magalhães, 2005)..	26
Figura 10: Imagem com ruído Gaussiano (a); imagem filtrada pela mediana 3x3 (b); e imagem filtrada pela mediana 5x5 (c) (Souza & Correia, 2007).....	26
Figura 11: Imagem Original (a); Imagem após a aplicação de um filtro passa-baixa (b); e após a aplicação de um filtro passa-alta (c) (C. A. R. Costa, 1998).	27
Figura 12: Exemplos de elementos estruturantes (Gonzalez; Woods, 2009).....	29
Figura 13: (a) Imagem original e (b) Imagem dilatada (Silva, 2023).....	30

LISTA DE QUADROS

Quadro 1: Análise de imagens de ovos do mosquito Anopheles realizada pelo algoritmo.....47

Quadro 2: Análise de imagens isoladas do ovo do mosquito Anopheles realizada pelo algoritmo.....48

LISTA DE SIGLAS E ABREVIATURAS

AM	Aprendizado de Máquina
CNN	<i>Convolutional Neural Network</i> (traduzido para redes neurais convolucionais)
FPR	<i>False Positive Rate</i> (traduzido para taxa de falsos positivos)
EE	Elemento Estruturante
L*a*b	Luminosidade, coordenada vermelho/verde, coordenada amarelo/azul
PDI	Processamento Digital de Imagens
RGB	<i>Red, Green e Blue</i> (traduzido para vermelho, verde e azul)
ROC	<i>Receiver Operating Characteristic</i> (traduzido para característica de operação do receptor)
TPR	<i>True Positive Rate</i> (traduzido para taxa de verdadeiros positivos)

SUMÁRIO

1. INTRODUÇÃO	13
1.1. OBJETIVOS	14
1.1.1. Objetivo Geral	14
1.1.2. Objetivos Específicos	14
1.2. DIVISÃO DO TRABALHO	15
2. REFERENCIAL TEÓRICO	17
2.1. PROCESSAMENTO DIGITAL DE IMAGENS	17
2.1.1. Imagens Digitais	18
2.1.2. Aquisição de Imagens	21
2.1.3. Pré-Processamento	22
2.1.3.1. Histograma.....	22
2.1.3.2. Filtragem.....	23
2.1.3.2.1. <i>Filtros passa-baixa</i>	24
2.1.3.2.2. <i>Filtro de Média</i>	24
2.1.3.2.3. <i>Filtro Gaussiano</i>	25
2.1.3.2.4. <i>Filtro de Mediana</i>	26
2.1.3.2.5. <i>Filtros passa-alta</i>	27
2.1.3.3. Morfologia Matemática	28
2.1.3.3.1. <i>Dilatação</i>	28
2.1.4. Segmentação	30
2.1.4.1. Limiarização de Otsu	31
2.1.4.2. Segmentação de bordas de <i>Canny</i>	32
2.1.5. Representação e Descrição	32
2.1.6. Reconhecimento e interpretação	33
2.2. LINGUAGEM PYTHON	33
2.3. APRENDIZADO DE MÁQUINA (AM).....	34
2.4. CURVA ROC.....	35
3. MATERIAIS E MÉTODOS	36
4. RESULTADOS E DISCUSSÃO	40
4.1. DESENVOLVIMENTO DO ALGORITMO	40
4.2. ANÁLISE DAS IMAGENS.....	43
4.3. COMPARAÇÃO DOS RESULTADOS OBTIDOS EM TRABALHOS SEMELHANTES FOCADOS NA DETECÇÃO DE OVOS DO <i>Aedes Aegypti</i>	50
5. CONSIDERAÇÕES FINAIS	52
5.1. TRABALHOS FUTUROS.....	53

REFERÊNCIAS	55
APÊNDICE A – ALGORITMO DESENVOLVIDO	60
ANEXO A – IMAGENS ANALIZADAS PELO ALGORITMO.....	62
ANEXO B – IMAGENS DE OVOS DO MOSQUITO ANOPHELES UTILIZADAS NO MACHINE LEARNING	72
ANEXO C – IMAGENS DE NÃO-OVOS UTILIZADAS NO MACHINE LEARNING.....	74

1. INTRODUÇÃO

Atualmente, a malária é um grave problema de saúde pública, afetando não somente o Brasil, mas também cerca de 84 países endêmicos e contabilizando cerca de 247 milhões de casos em 2021 (WHO, 2022). Segundo a Fundação Oswaldo Cruz (Fiocruz), é uma doença infecciosa, febril, aguda e potencialmente grave, sendo transmitida ao homem na maioria das vezes através da picada de mosquitos do gênero *Anopheles*, também conhecidos popularmente como mosquitos-prego, quando infectados por um parasita do gênero *Plasmodium*. Por essa razão, se faz extremamente necessário que medidas de prevenção sejam adotadas, uma vez que ainda não está disponível em nosso país uma vacina que seja totalmente eficaz contra a doença.

Tais medidas de prevenção dizem respeito principalmente ao controle do mosquito transmissor da doença, impedindo sua reprodução e proliferação. Dentre as ações mais comuns adotadas atualmente, temos o uso de redes mosquiteiras ou repelentes para a prevenção de picadas, a utilização de inseticidas e o escoamento de águas paradas. (Caraballo & King, 2014).

Esse trabalho propõe o desenvolvimento de um sistema computacional capaz de auxiliar nas medidas de prevenção, identificando a presença de ovos do mosquito *Anopheles* e realizando sua contagem, a fim de avisar o usuário quanto à possibilidade de contaminação eminente para que ele possa adotar as ações necessárias para se proteger de uma possível infecção.

Assim, para construção desse Trabalho de Conclusão de Curso (TCC), partimos da seguinte questão norteadora: É possível desenvolver um aplicativo computacional capaz de detectar a presença de ovos do mosquito transmissor da malária (*Anopheles*) e realizar sua contagem, de forma a alertar os usuários da região com relação a possibilidade de contaminação pela doença?

O desenvolvimento de um algoritmo que seja capaz de realizar a identificação e a contagem de ovos do mosquito *Anopheles* se torna possível a partir de técnicas de detecção de bordas, como o método de *Canny* por exemplo, propostas ao estudarmos o processamento digital de imagens e através de conceitos presentes na inteligência artificial, como o *Machine Learning*. Já o programa em si pode ser elaborado por meio da linguagem de programação *Python* ou através do software *Matlab*, baseado na linguagem C. Assim, optamos por utilizar a linguagem *Python*

como ferramenta para desenvolver o algoritmo proposto devido a uma maior familiaridade com as bibliotecas disponíveis.

1.1. OBJETIVOS

1.1.1. Objetivo Geral

Partindo dessas premissas, o presente estudo tem como objetivo geral desenvolver um algoritmo computacional capaz de identificar e contar ovos do mosquito transmissor da malária (*Anopheles*) a fim de ser utilizado em locais cuja presença de água parada possa se tornar um risco para a transmissão e proliferação da doença.

1.1.2. Objetivos Específicos

Desse objetivo geral decorrem alguns específicos, dos quais podemos citar:

- a) Realizar o estudo de técnicas pertencentes a disciplina de Processamento Digital de Imagens e selecionar as mais adequadas, a fim de serem implementadas no algoritmo;
- b) Observar e estudar as características presentes nas imagens do banco de dados disponível;
- c) Desenvolver um algoritmo em *Python* que identifique os padrões presentes nas imagens e as compare através de *machine learning*, indicando a presença ou não de ovos do mosquito *Anopheles*;
- d) Implementar no mesmo algoritmo um código capaz de realizar a contagem dos ovos, mostrando ao final quantos estão presentes nas imagens obtidas;
- e) Verificar qual a porcentagem da taxa de erro presente no algoritmo desenvolvido tanto para os falsos negativos (ovos que estejam na imagem, mas não estejam sendo contabilizados) quanto para os falsos positivos (outros objetos presentes nas imagens que estejam sendo contabilizados como ovos), comparando com o resultado obtido em

trabalhos semelhantes relacionados aos ovos do *Aedes Aegypti* (mosquito da dengue).

O tema justifica-se tanto pela relevância social quanto pela contribuição acadêmica que pode oferecer. Socialmente, a malária representa uma das principais causas de morbidade e mortalidade em muitas regiões tropicais e subtropicais do mundo, com um impacto significativo na qualidade de vida, produtividade econômica e desenvolvimento social. Um sistema computacional que detecte e conte ovos do mosquito *Anopheles* pode agir como uma ferramenta de vigilância ambiental, oferecendo uma resposta precoce e precisa à presença do vetor da malária. Com essa informação, comunidades podem implementar medidas de controle de vetores de maneira mais eficaz, reduzindo assim a incidência da doença.

1.2. DIVISÃO DO TRABALHO

A presente monografia apresentada a seguir está formatada da seguinte forma:

O primeiro capítulo deste TCC apresenta essa própria introdução ao trabalho, no qual fundamenta-se o tema, delimitação do tema, problema, hipóteses, objetivos (geral e específico) e justificativa.

O segundo capítulo apresenta o referencial teórico que foi utilizado no desenvolvimento dessa pesquisa, abordando os alicerces do Processamento Digital de Imagens, que inclui tópicos como a natureza de imagens digitais, métodos de aquisição, e técnicas de pré-processamento como ajustes de histograma, filtragem (passa-baixa e passa-alta) e morfologia matemática. Avança para as estratégias de segmentação, com especial atenção à Limiarização de Otsu e à detecção de bordas de *Canny*, e discute a representação e descrição de características para posterior reconhecimento e interpretação. Explora-se ainda a linguagem *Python* e Aprendizado de Máquina (AM), e conclui-se com a importância da Curva *ROC* na avaliação do desempenho dos algoritmos de classificação, estabelecendo assim a fundação teórica sobre a qual a pesquisa e desenvolvimento do aplicativo são construídos.

O terceiro capítulo evidencia os materiais e métodos utilizados na pesquisa. Nele, são descritos os passos metodológicos, desde a configuração inicial do ambiente de desenvolvimento no *Google Colab* até a implementação e avaliação do algoritmo proposto. O capítulo inicia com a seleção e preparação do conjunto de imagens digitais,

detalhando o processo de coleta, as características das imagens e a importância de cada aspecto no contexto da detecção e contagem de ovos de mosquito *Anopheles*.

O quarto capítulo mostra os principais resultados e discussão sobre o desenvolvimento e eficácia do algoritmo proposto, começando com o desenvolvimento do algoritmo. Seguindo pela análise das imagens e a comparação feita com estudos anteriores voltados para a detecção de ovos do *Aedes Aegypti*, avaliando como o algoritmo se alinha aos resultados prévios.

E por fim, as considerações finais que descreve se os principais objetivos foram alcançados e evidenciam as propostas de estudos futuros, com sugestões de aperfeiçoamentos para trabalhos futuros e propostas a serem implementadas através do programa desenvolvido.

2. REFERENCIAL TEÓRICO

2.1. PROCESSAMENTO DIGITAL DE IMAGENS

Segundo Gonzales & Woods (2010), a área de processamento digital de imagens envolve processos cujas entradas e saídas são imagens e, além disso, envolve processo de extração de atributos, inclusive o reconhecimento de objetos individuais. O processamento de imagens é composto por diversas fases, que englobam uma série de técnicas computacionais capazes de realizar operações para melhorar a visualização de características presentes nas imagens, de modo que o olho humano seja incapaz de determinar. (MARQUES, 2018)

As etapas fundamentais de um sistema de processamento digital de imagens incluem: domínio do problema, aquisição da imagem, pré-processamento, segmentação, representação e descrição, reconhecimento e interpretação do resultado (Figura 1).

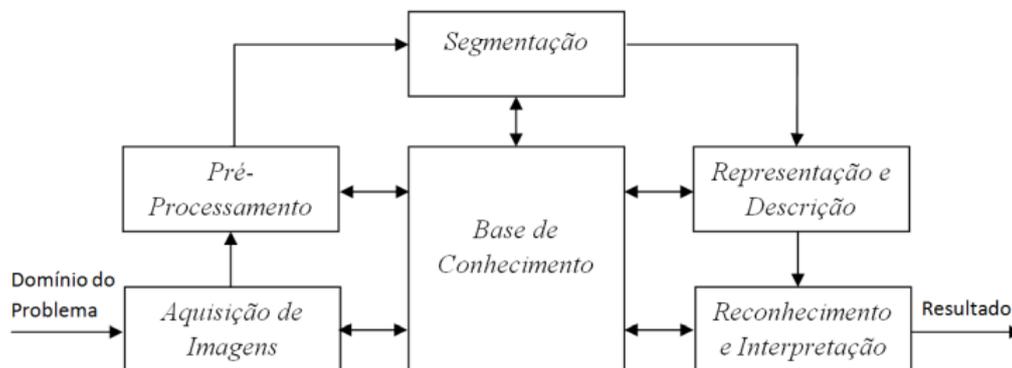


Figura 1: Etapas fundamentais do Processamento Digital de Imagens (Gonzales & Woods, 2000).

Conforme Scuri (1999), as imagens digitais são compostas por uma matriz de pixels (elementos de imagem) que representam o valor da intensidade de cor em cada ponto da imagem. Assim, de acordo com Lacerda (2015), o processamento digital de imagens envolve várias etapas, incluindo:

- a) A aquisição da imagem, que pode ser realizada por meio de microscópios ópticos e eletrônicos, câmeras digitais, scanners e outras fontes de imagens digitais;

- b) O pré-processamento, que é uma etapa importante que envolve o melhoramento da imagem adquirida ou o realce de determinadas características através da remoção de ruído, da correção de distorções, do ajuste da escala de cinza, entre outras técnicas;
- c) O processamento principal, que envolve a aplicação de algoritmos e técnicas matemáticas para realizar tarefas específicas, como detecção de bordas, segmentação de imagem, filtragem e reconhecimento de padrões. A detecção de bordas, por exemplo, é uma técnica utilizada para identificar mudanças abruptas na intensidade de cor na imagem, o que pode ajudar a detectar as linhas de determinado objeto ou identificar regiões de interesse;
- d) A segmentação de imagem é uma técnica utilizada para dividir uma imagem em regiões com características semelhantes. Isso pode ser útil para separar objetos em uma imagem ou identificar regiões de interesse. Já a filtragem é uma técnica utilizada para remover o ruído de uma imagem ou suavizá-la. O reconhecimento de padrões, por sua vez, é uma técnica utilizada para identificar objetos em uma imagem ou classificá-la em diferentes categorias (Saúde, 2019);
- e) Por fim, o pós-processamento envolve a análise dos resultados do processamento e a tomada de decisões com base nesses resultados. As aplicações do processamento digital de imagens são vastas e incluem diagnóstico médico, reconhecimento de padrões, reconhecimento de caracteres, processamento de vídeo, análise de imagens de satélite, análise de imagens de segurança, entre outras (Saúde, 2019).

Portanto, podemos concluir que o processamento digital de imagens é uma área importante da engenharia elétrica e da computação que permite a manipulação de imagens digitais para extrair informações úteis para uma ampla variedade de aplicações.

2.1.1. Imagens Digitais

Uma imagem digital pode ser representada como uma matriz bidimensional contendo números binários de modo a permitir seu armazenamento, transferência, impressão ou reprodução, e seu processamento por meios eletrônicos. No

processamento digital de imagens, a imagem representada por uma matriz é constituída por valores em função da luz incidente sobre os objetos presentes na cena da imagem. Essa função pode ser denominada pixel (*picture elements*) e define cor, textura, dimensão e posição dos elementos da imagem. (E. O. Jesus & Costa Jr, 2015).

Nesse sentido, cada pixel contém informações sobre sua posição na imagem e sua cor. A cor de um pixel é definida por sua intensidade em três canais de cor: vermelho, verde e azul (RGB) (Burger & Burge, 2008). Este modelo descreve as cores como resultado da adição das três cores primárias, com uma intensidade que pode variar entre 0 e 1. O valor 1 corresponde à intensidade máxima que a cor pode ser apresentada e o valor 0 à intensidade mínima (JUNIOR, 2009).

Ainda segundo JUNIOR (2009), tradicionalmente, as implementações do modelo RGB nos sistemas gráficos empregam valores inteiros entre 0 e 255 para exprimir o valor da intensidade de cada componente no lugar de valores entre 0 e 1.

O formato do arquivo é a forma como a imagem será armazenada, isto é, guardada ou salva em meio digital. Existem inúmeros formatos referentes a tais imagens, porém os mais utilizados são o *JPEG* (ou *.jpg*) e o *TIFF* (ou *.tif*) (Machado & Souki, 2004). Além disso, as imagens digitais podem ser manipuladas e editadas usando um software de edição de imagem, como o *Adobe Photoshop*, que permite que o usuário altere a cor, brilho, contraste e outros aspectos, bem como adicionar ou remover elementos da imagem (Solomon & Breckon, 2011).

Em suma, as imagens digitais são compostas por pixels, que são pequenos pontos coloridos que se unem para formar uma imagem completa. Cada pixel contém informações sobre sua posição na imagem e sua cor, e as imagens armazenadas em formato digital podem também ser manipuladas usando software de edição de imagem.

Podemos dizer, portanto, que imagem digital é a representação de uma cena por meio de um conjunto de elementos discretos e de tamanhos finitos, denominados *pixels*, quando colocados em um arranjo bidimensional. (Martins, n.d.).

Dessa forma, podemos definir uma imagem monocromática como sendo uma função da intensidade luminosa, sendo seu valor, em qualquer ponto do plano (x,y) proporcional ao nível de cinza (ou ao brilho) naquele ponto. A figura 2 mostra uma imagem monocromática e a convenção utilizada para o par de eixos (x,y) nesse trabalho.



Figura 2: Uma imagem monocromática e a convenção utilizada para o par de eixos (x,y) . (Marques Filho & Vieira Neto, 1999).

Uma imagem colorida seguindo o modelo de cores RGB, por exemplo, baseia-se em um sistema de coordenadas cartesianas, em que o espaço de cores é um cubo (Pedrini & Schwartz, 2008), conforme podemos verificar na figura 3.

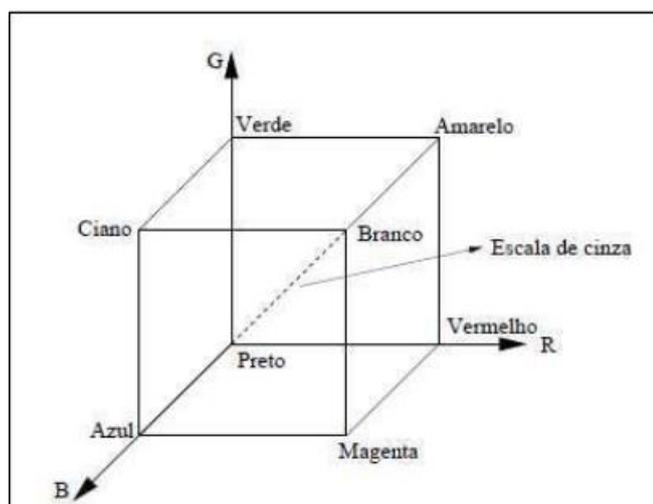


Figura 3: Modelo RGB (Pedrini & Schwartz, 2008).

Analisando o espaço de cores do modelo RGB, observa-se que os valores primários estão representados nos três vértices do cubo, enquanto as cores secundárias como o magenta, o ciano e o amarelo estão representados nos outros vértices (Lopes et al., 2013). A escala de cinza nesse plano se estende do preto até o branco, ao longo do segmento que une essas duas cores. A figura 4 apresenta um outro esquema com as cores do modelo RGB.

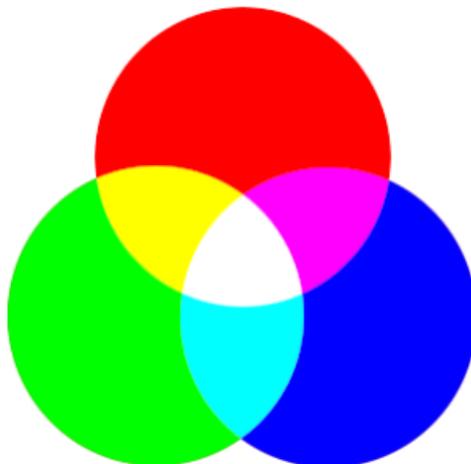


Figura 4: Círculos cromáticos RGB (Rocha, 2010).

Por fim, na figura 5 podem ser vistas duas representações de uma mesma imagem: em cores (modelo RGB) e em escala de cinza.



Figura 5: Representações de uma imagem: (a) cores (formato RGB) e (b) escala de cinza (Peixoto, 2010).

Neste trabalho serão utilizadas imagens representadas em escala de cinza, sendo que faremos a conversão das imagens obtidas no modelo de cores RGB.

2.1.2. Aquisição de Imagens

Segundo Marques Filho & Vieira Neto (1999), a aquisição de imagens digitais tem como função converter uma cena real tridimensional em uma imagem analógica. Esta etapa compreende dois elementos, sendo que o primeiro consiste em um dispositivo físico que produz um sinal elétrico de saída proporcional a um nível de energia percebido, enquanto o segundo, chamado digitalizador, é um dispositivo para a conversão da saída elétrica analógica em sinal digital (Gonzales & Woods,

2000). A saída da etapa de aquisição consiste, portanto, na representação da imagem digital obtida pelo dispositivo.

Neste trabalho, serão utilizadas imagens provenientes de um banco de dados e em grande parte capturadas a partir de um microscópio óptico, de forma que os ovos do mosquito *Anopheles* possam ser visualizados e identificados por nosso algoritmo.

2.1.3. Pré-Processamento

Segundo Souza & Correia (2007), a etapa de pré-processamento consiste na aplicação de técnicas de realce de imagens, destacando uma região dentro da imagem e permitindo sua visualização com mais detalhes, de forma que o resultado dessa operação seja mais apropriado para a aplicação específica do que a imagem original.

Estas técnicas envolvem duas categorias principais: métodos que operam no domínio espacial e métodos que operam no domínio da frequência. Técnicas de processamento no domínio espacial baseiam-se em filtros que manipulam o plano da imagem, enquanto técnicas de processamento no domínio da frequência se baseiam em filtros que agem sobre o espectro da imagem. (Esquef et al., 2003)

No nosso trabalho, a etapa de pré-processamento consiste em realizar o tratamento inicial das imagens obtidas, convertendo-as do sistema RGB para a escala de cinza e obtendo o seu respectivo histograma, de forma que nosso algoritmo possa processá-las com clareza, identificar os ovos do mosquito da malária presentes em cada uma delas e, por fim, realizar sua contagem.

2.1.3.1. Histograma

Segundo Queiroz & Gomes (2001), o histograma de uma imagem traduz a distribuição estatística dos seus níveis de cinza. Trata-se, portanto, de uma representação gráfica do número de pixels associado a cada nível de cinza presente na imagem, podendo também ser expressa em termos do percentual do número total de pixels na imagem. Através da visualização do histograma de uma imagem, obtemos uma indicação de sua qualidade quanto ao nível de contraste e quanto ao

seu brilho médio (Neves & Pelaes, 2001). Cada elemento desse conjunto é calculado como:

$$P_r(r_k) = \frac{n_k}{n}$$

Onde $k = 0, 1, 2, \dots, L-1$ em que L é o número de níveis de cinza da imagem; n corresponde ao número total de pixels na imagem; n_k corresponde ao número de pixels cujo nível de cinza corresponde a k ; e $P_r(r_k)$ é a probabilidade do k -ésimo nível de cinza.

A figura 6 representa o histograma de uma imagem 3x5 de acordo com a definição anterior:

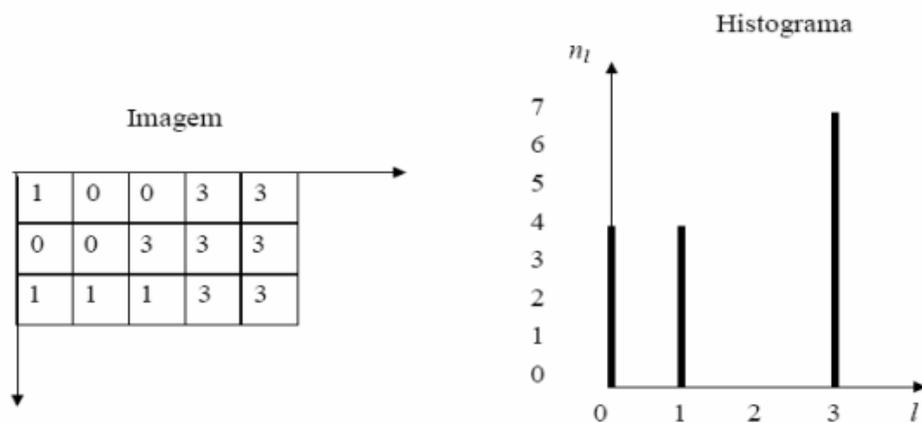


Figura 6: Uma imagem 3x5 e seu respectivo histograma (Souza & Correia, 2007).

2.1.3.2. Filtragem

Segundo Batista (2005), os filtros podem ser definidos como operadores unitários que processam uma imagem de entrada produzindo assim uma imagem de saída. Matematicamente, podemos definir esse processo de filtragem com a seguinte equação, em que uma função T mapeia uma imagem de entrada f gerando, por fim, a imagem de saída g .

$$g = T[f]$$

O princípio básico de funcionamento dos filtros é a convolução de uma imagem f com máscaras de deslocamento. Uma máscara com centro em (x,y) é colocado sobre um pixel da imagem de entrada de tal forma que esse pixel passa a ser substituído por um valor que não somente considera os pixels vizinhos, mas também os pesos da máscara. A soma de todos os produtos dos pesos da máscara

pelos pixels correspondentes na imagem resulta, por fim, em um novo valor de nível de cinza que irá substituir o pixel central (CARVALHO, 2003).

Segundo Oliveira et al. (2004), os filtros mais utilizados em processamento digital de imagens para a redução de ruído são os passa-baixa de média, gaussiana e de mediana. Podemos citar ainda o filtro passa-alta como sendo bastante utilizado para aplicações em geral.

2.1.3.2.1. Filtros passa-baixa

Ainda segundo Oliveira et al. (2004), os filtros passa-baixa são formados por *kernels* contendo pesos positivos e privilegiam a passagem de componentes de baixa frequência, atenuando assim os componentes de alta frequência, como bordas e ruídos.

2.1.3.2.2. Filtro de Média

É um dos tipos mais simples de filtro passa-baixa existentes e seu efeito é substituir o nível de cinza do pixel pela sua média aritmética e de seus vizinhos (Crósta, 1992). A figura 7 ilustra um exemplo de filtro passa-baixa de média 3x3, 5x5 e 7x7, enquanto a figura 8 apresenta o resultado desses filtros quando aplicados a uma imagem de entrada (a), sendo as figuras 8-c e 8-d o resultado da aplicação dos filtros 3x3 e 5x5, respectivamente.

Podemos notar ao comparar as imagens filtradas que as figuras 8-c e 8-d tiveram o ruído reduzido e, com isso, ficaram mais borradas e menos nítidas, o que indica que nem sempre esse tipo de filtro é o mais adequado.

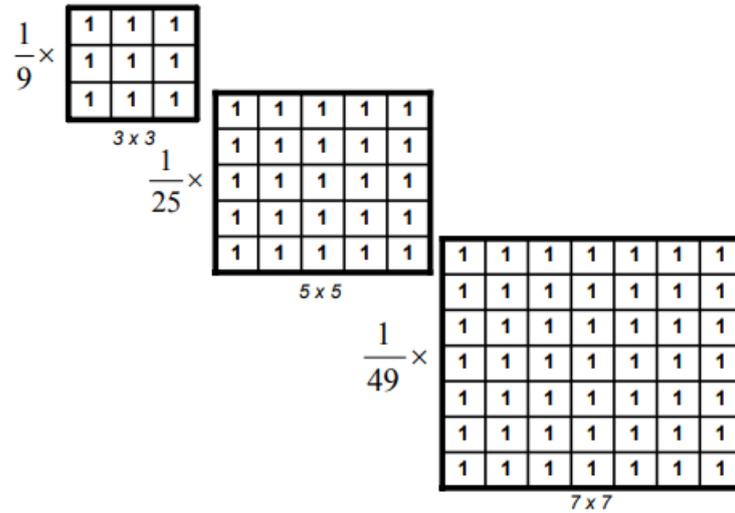


Figura 7: Filtros passa-baixa de média do tipo 3x3, 5x5 e 7x7 (Oliveira et al., 2004).



Figura 8: Aplicação do filtro de média: (a) imagem original; (b) imagem com ruídos atenuados; (c) imagem filtrada com uma máscara 3x3 e (d) imagem filtrada com uma máscara 5x5 (Queiroz & Gomes, 2001).

2.1.3.2.3. Filtro Gaussiano

O filtro Gaussiano, por sua vez, aplica a cada *pixel* uma média ponderada de seus vizinhos, na qual os pesos são atribuídos em função da distância de cada *pixel* para o pixel central. Dessa forma, o resultado representa uma suavização na imagem (Jesus et al., 2015). Esse tipo de filtro é o mais adequado em aplicações de reconhecimento facial e de objetos, uma vez que utiliza todos os pixels presentes em uma imagem e não considera os ruídos (Haddad & Akansu, 1991).

A figura 9 apresenta uma imagem original seguida do efeito após a aplicação do filtro Gaussiano, servindo para exemplificar o uso desse filtro.



Figura 9: Aplicação do filtro Gaussiano: (a) imagem original e (b) imagem após a aplicação de um filtro Gaussiano (Oliveira & Magalhães, 2005).

Por se tratar do tipo de filtro mais adequado para aplicações de reconhecimento facial e detecção de objetos, suavizando a imagem sem levar em conta os ruídos, utilizaremos em nosso trabalho um filtro Gaussiano para que o algoritmo detecte com mais clareza os ovos do mosquito *Anopheles*.

2.1.3.2.4. Filtro de Mediana

Segundo Souza & Correia (2007), em uma filtragem por mediana, o nível de cinza de cada pixel é substituído pela mediana dos níveis de cinza dos pixels de sua vizinhança. A mediana de um conjunto de pixels é representada pelo valor central, se o número de pixels total da imagem for ímpar, e é formado pela média aritmética dos dois pixels situados na posição central da imagem, se o número total for par.

A figura 10 apresenta uma imagem original (a) quando aplicados os filtros de mediana 3x3 (b) e 5x5 (c).



Figura 10: Imagem com ruído Gaussiano (a); imagem filtrada pela mediana 3x3 (b); e imagem filtrada pela mediana 5x5 (c) (Souza & Correia, 2007).

2.1.3.2.5. *Filtros passa-alta*

Os filtros passa-alta, ao contrário dos filtros passa-baixa, realçam detalhes das imagens como bordas, linhas, curvas, limites entre áreas e inclusive, ruídos (Silva et al., 2015). Segundo Crósta (1992), considerando que uma imagem é formada por componentes de alta e baixa frequência, a maneira mais simples de aplicar uma filtragem passa-alta, é subtraindo a imagem suavizada (passa-baixa) da imagem original.

Para que possamos fazer uma melhor comparação, a figura 11 apresenta uma imagem original (a) após serem aplicados os filtros passa-baixa (b) e passa-alta (c).

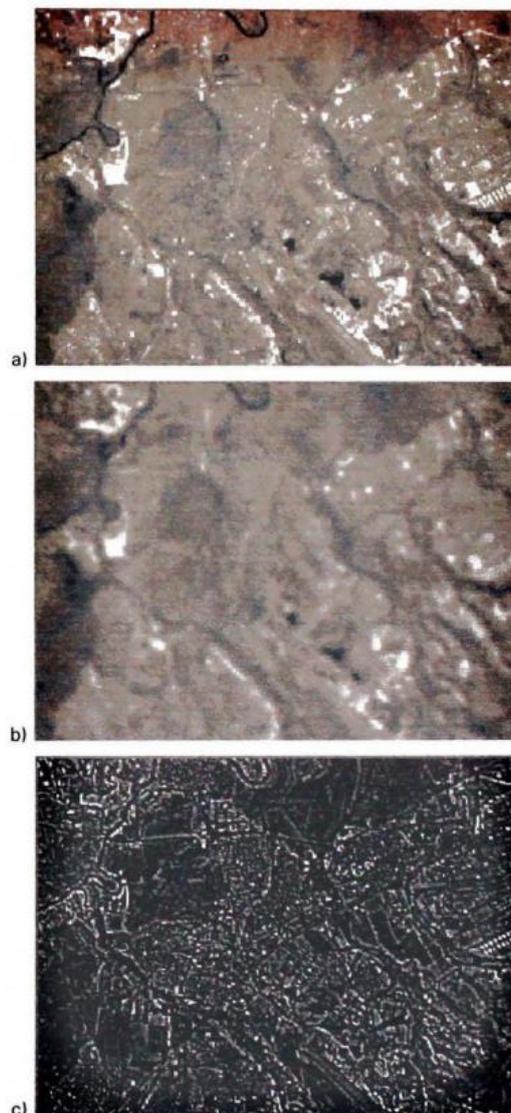


Figura 11: Imagem Original (a); Imagem após a aplicação de um filtro passa-baixa (b); e após a aplicação de um filtro passa-alta (c) (C. A. R. Costa, 1998).

2.1.3.3. Morfologia Matemática

A morfologia matemática aborda uma imagem como se fosse um conjunto e emprega um conjunto menor, denominado elemento estruturante, para explorá-la. Esta abordagem, que se baseia na teoria dos conjuntos, torna a morfologia matemática particularmente apropriada para o tratamento de informações visuais (Yu et al., 2008).

De acordo com Sousa et al. (2020), originalmente concebidas para imagens binárias, as operações da morfologia matemática têm sua teoria fundamental desenvolvida nesse contexto. A morfologia, aplicável em uma ampla gama de processos de tratamento de imagens, incluindo realce, filtragem, segmentação e esqueletização - que diz respeito à forma e estrutura de um objeto, bem como à disposição e conexão entre as partes desse objeto.

As operações morfológicas são executadas mediante um elemento estruturante pré-definido, cuja geometria é essencial na definição do efeito da operação sobre a imagem. Este elemento estruturante, que pode assumir diversas formas como discos, quadrados ou formas personalizadas, é empregado para sondar a imagem de entrada, aplicando regras específicas de inclusão ou exclusão de pixels (Sousa et al., 2020).

Além do elemento estruturante, a erosão e a dilatação constituem as operações fundamentais da Morfologia Matemática. Estas duas técnicas básicas, ao serem combinadas, permitem a criação de uma variedade de operadores complexos. A erosão funciona através da remoção de partes da imagem utilizando o elemento estruturante, um processo que pode ser efetuado com a ajuda de uma função específica do *Matlab*, que leva em consideração tanto o elemento estruturante quanto a imagem binarizada ou em escala de cinza. Já a dilatação, em contrapartida, pode ser entendida como a adição do elemento estruturante à imagem. Desta forma, erosão e dilatação são operações inversas uma da outra.

2.1.3.3.1. Dilatação

A dilatação tem como objetivo principal aumentar regiões de objetos em uma imagem, geralmente aplicada em imagens binárias, ou seja, imagens que possuem dois níveis de intensidade: preto e branco, representando o fundo e os objetos,

respectivamente (Virgílio, 2021) Nesse sentido a dilatação é realizada da seguinte forma:

- a) Definição do Elemento Estruturante (EE) conforme a Figura 12: Para realizar a dilatação, primeiro define-se um elemento estruturante, que é uma pequena forma ou máscara (como um disco, quadrado, cruz etc.) que determina a forma como a dilatação é aplicada (Gonzalez; Woods, 2009).

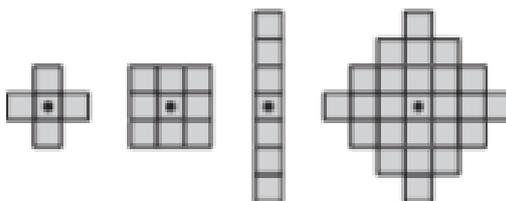


Figura 12: Exemplos de elementos estruturantes (Gonzalez; Woods, 2009).

- b) Posicionamento do EE: O elemento estruturante é posicionado sobre cada pixel da imagem, geralmente começando pelo canto superior esquerdo e prosseguindo até cobrir toda a imagem (Gonzalez; Woods, 2009).
- c) Aplicação da Dilatação: Para cada posicionamento do elemento estruturante, verifica-se se ao menos um pixel sob o EE é um *pixel* de objeto (geralmente representado pelo valor 1 em imagens binárias). Se sim, o *pixel* central do EE é configurado como pixel de objeto na imagem de saída. Logo, essa operação morfológica aumenta os entornos da imagem, definida pela equação:

$$A \oplus B = \{z | ((\hat{B})_z \cap A \neq \emptyset)\}$$

- d) Resultado da dilatação: O resultado é uma imagem onde as regiões de objetos são expandidas, conforme a Figura 13. As fronteiras dos objetos se movem para fora, o que pode ser útil para fechar pequenos buracos e espaços dentro dos objetos ou para conectar regiões próximas (Silva, 2023).

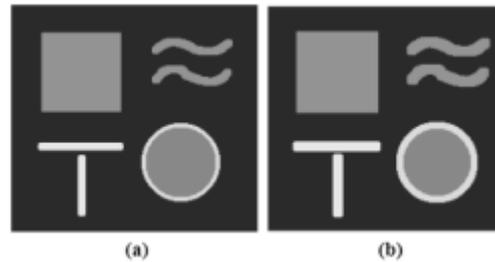


Figura 13: (a) Imagem original e (b) Imagem dilatada (Silva, 2023).

Deste modo, Silva (2023) discorre que, as aplicações da dilatação são realizadas para a melhoria da visualização, em imagens de baixa resolução ou com objetos muito pequenos, a dilatação pode ajudar a tornar esses objetos mais visíveis. E ainda no processamento de imagens médicas para destacar características importantes em imagens médicas, como aumentar a visibilidade de tumores ou outras estruturas anatômicas.

A dilatação, conforme Vigilio (2021) ao expandir os pixels de objetos sobre seus vizinhos com base em um elemento estruturante definido, é uma ferramenta poderosa que ajuda a simplificar e destacar características importantes em uma variedade de aplicações de processamento de imagens. É importante, contudo, utilizar a dilatação com cuidado, pois a expansão indiscriminada de objetos pode levar à perda de detalhes finos e à sobreposição de objetos anteriormente separados.

2.1.4. Segmentação

A segmentação de imagens é um processo que consiste na triagem ou identificação dos objetos contidos na imagem, separando-a em diferentes regiões. O objetivo da segmentação é permitir uma análise independente de cada uma dessas regiões. A segmentação é realizada por meio do agrupamento de pixels com características ou propriedades semelhantes, como intensidade e textura, ou através da eliminação de características específicas para destacar a diferença entre as regiões segmentadas (A. M. P. G. Vale, 2014). Em resumo, a segmentação de imagens é uma técnica que permite a separação de uma imagem em regiões distintas com o objetivo de facilitar sua análise.

Logo, será utilizada em nosso trabalho a segmentação por descontinuidade através da detecção de bordas do objeto de interesse, separando-as do resto da imagem

em si e comparando-as com o formato dos ovos utilizados nos modelos dispostos para o treinamento do algoritmo.

2.1.4.1. Limiarização de Otsu

A limiarização de Otsu é uma técnica de segmentação de imagens amplamente utilizada em PDI. Essa técnica se baseia no cálculo do valor de limiar ideal que separa os pixels da imagem em duas classes distintas: uma classe correspondente ao objeto de interesse e outra classe correspondente ao fundo (Bertholdo, 2007).

O processo de limiarização de Otsu envolve os seguintes passos de acordo com Bertholdo (2007):

- a) Cálculo do histograma da imagem: o histograma é calculado contando o número de pixels que possuem cada valor de intensidade.
- b) Cálculo da probabilidade de cada valor de intensidade: a partir do histograma, é possível calcular a probabilidade de cada valor de intensidade, dividindo o número de pixels com o valor de intensidade correspondente pelo número total de pixels na imagem.
- c) Cálculo da média global da intensidade: a média global da intensidade é continuada a partir da soma do produto de cada valor de intensidade pela sua probabilidade.
- d) Cálculo da variância interclasse: a variância interclasse é uma medida da separabilidade das duas classes de pixels. Ela é comutada para cada valor possível de limiar, a partir da fórmula:

$$\sigma^2 = w_1 * w_2 * (\mu_1 - \mu_2)^2$$

onde:

- w_1 e w_2 são as probabilidades das duas classes de pixels;
 - μ_1 e μ_2 são as médias das intensidades das duas classes.
- e) Determinação do valor de limiar ideal: o valor de limiar ideal é aquele que maximiza a variância interclasse, ou seja, o valor que resulta na maior separabilidade possível entre as duas classes de pixels.
 - f) Segmentação da imagem: após determinado o valor de limiar ideal, a imagem é segmentada em duas classes de pixels, com base no valor de intensidade de cada *pixel* em relação ao valor de limiar.

A limiarização de Otsu é uma técnica simples e eficaz para segmentação de imagem, especialmente quando não há informações prévias sobre a imagem ou quando a imagem apresenta um alto grau de variação na intensidade dos pixels. No entanto, é importante destacar que essa técnica pode não ser adequada para imagens com ruído ou com baixo contraste, pois nesses casos o cálculo do valor de limiar ideal pode ser afetado por efeitos sonoros.

2.1.4.2. Segmentação de bordas de *Canny*

A técnica de segmentação de bordas de *Canny*, desenvolvida em 1986 por John F. Canny, compreende um algoritmo com múltiplos processos para a detecção de bordas de um ou mais objetos em uma imagem, tais como o processo da detecção, da localização e da resposta múltipla (G. Do Vale & Poz, 2004).

Ainda segundo G. Do Vale & Poz (2004), o processo de detecção corresponde a capacidade do algoritmo de capturar com uma baixa taxa de erro as bordas “verdadeiras” da imagem, ou seja, aquelas que realmente correspondem ao objeto em si. Já o processo da localização estabelece que a distância entre os pontos detectados e o centro verdadeiro da borda deve ser minimizada, de forma que a localização dos objetos esteja próxima da representada na imagem original. Por fim, o critério da resposta múltipla estabelece que um algoritmo detector de bordas ótimo deve obter uma única resposta por borda, evitando por exemplo, detectar mais de uma borda onde há apenas uma.

2.1.5. Representação e Descrição

Segundo Costa (2017), a saída do estado de segmentação é constituída principalmente por dados em forma de pixels, de forma que se faz necessário uma conversão para um formato mais adequado ao processamento computacional. Assim, é realizada a representação apropriada do objeto para a posterior descrição dos dados.

A descrição, por sua vez, é muitas vezes apenas um conjunto de dados que são chamados descritores do objeto, de acordo com Vale (2014). Este processo, também denominado de extração de características, procura extrair propriedades que resultem em alguma informação de interesse ou que sejam básicas para a distinção dos objetos.

No nosso trabalho, essa etapa compreende a separação dos objetos para o reconhecimento posterior, bem como a extração de características como o comprimento e a área dos ovos, permitindo uma melhor identificação deles por nosso algoritmo.

2.1.6. Reconhecimento e interpretação

Por fim, o último estágio no processamento de imagens digitais envolve o reconhecimento e a interpretação do objeto por meio do algoritmo. Segundo Braga & Cardoso (2010), o reconhecimento consiste em atribuir determinado identificador a uma informação e a interpretação compreende na atribuição de significado a um conjunto de dados reconhecidos.

A identificação de um caractere qualquer em meio a outros, por exemplo, requer a associação de suas descrições ao rotulo do caractere correspondente para que seu reconhecimento possa ser feito corretamente. Já a interpretação procura atribuir significado a um conjunto de entidades rotuladas, como três caracteres seguidos por um hífen e mais quatro números, que podem ser interpretados em um sistema como a placa de um veículo por exemplo (Estigoni, 2006).

Em nosso trabalho, essa etapa corresponde ao reconhecimento do ovo do mosquito em si e de alguns negativos através das características atribuídas a eles, identificando-os e realizando a contagem dos ovos positivos em seguida.

2.2. LINGUAGEM PYTHON

Segundo Borges (2014), a linguagem *Python* foi desenvolvida inicialmente em 1990 por Guido van Rossum, a partir de outra linguagem de programação existente na época, denominada ABC.

Manzano (2018) define o *python* como uma linguagem de alto nível interpretada, o que significa que os algoritmos criados na forma de scripts são executados dentro do próprio programa sem que sejam compilados. A linguagem também aceita o paradigma de programação orientada a objetos por padrão e, devido à simplicidade na sua forma de escrita, aceita programação nos estilos estruturada (procedural/imperativo) e funcional.

É importante também ter em mente ao programar que se trata de uma linguagem *case sensitive*, ou seja diferencia caracteres minúsculos dos maiúsculos tanto na

execução de determinado comando quanto na declaração de variáveis ('A' e 'a' são variáveis diferentes, por exemplo) (Guedes & Bertol, 2018).

Dois aspectos relacionados a tipagem de dados também merecem destaque se tratando de *python*: a tipagem dinâmica e a tipagem forte.

Segundo Coelho (2007), tipagem automática refere-se ao tipo da variável que pode ser definida automaticamente a partir do algoritmo desenvolvido e tipagem forte diz respeito ao fato de não haver possibilidade de conversão automática de tipos de variável, sendo que uma *string* 123, por exemplo, não pode ser somada com o número inteiro 3.

A linguagem *python* pode ter diferentes utilizações como o processamento de imagens, ciência de dados, inteligência artificial, automação, aplicações envolvendo *Internet of Things* etc. Por se tratar de uma linguagem simples de se programar, porém completa, foi a linguagem que escolhemos para desenvolver nosso algoritmo.

2.3. APRENDIZADO DE MÁQUINA (AM)

Segundo Monard & Baranauskas (2003), o Aprendizado de Máquina (AM, ou *Machine Learning – ML*, em inglês) é uma área da inteligência artificial que objetiva o desenvolvimento de técnicas computacionais e a construção de sistemas automatizados capazes de adquirir conhecimento sem a necessidade de auxílio humano direto. De acordo com essa mesma definição, Mitchell (1997) definiu a área de estudo como “um campo que dá ao computador a capacidade de aprender sem ser explicitamente programado para tal”.

Segundo Russell & Norvig (2010), as tarefas que envolvem AM podem ser classificadas em três categorias, sendo elas o *aprendizado supervisionado*, o *não-supervisionado* e o *aprendizado por reforço*.

Paixão et al. (2022) definiu o aprendizado supervisionado como sendo o baseado no treinamento de um sistema com base em uma classificação já atribuída anteriormente pelo programador, enquanto aprendizado não-supervisionado se refere a capacidade de atribuir e classificar informações sem a necessidade de uma classificação atribuída de forma prévia.

Por fim, aprendizado por reforço pode ser definido como aquele em que um sistema aprende de forma dinâmica a tomar decisões de acordo com o ambiente que o cerca, não tendo assim necessidade de um ensinamento prévio (Bianchi, 2004).

No nosso projeto, utilizaremos a técnica da aprendizagem supervisionada como forma de treinar nosso algoritmo, ensinando-o através de figuras isoladas o que pode ser definido como ovo do mosquito *Anopheles* (positivo) e o que pode ser definido como outro objeto semelhante (negativo).

2.4. CURVA ROC

A Curva Característica de Operação do Receptor, conhecida como Curva *ROC* (do inglês *Receiver Operating Characteristic*), é uma ferramenta de análise gráfica utilizada para avaliar o desempenho de sistemas de classificação binária. A Curva *ROC* é gerada ao se plotar a taxa de verdadeiros positivos (*TPR* - *True Positive Rate*) contra a taxa de falsos positivos (*FPR* - *False Positive Rate*) em diferentes limiares de classificação. A *TPR* é também conhecida como sensibilidade, enquanto a *FPR* é 1 menos a especificidade. Vejamos:

- a) Taxa de Verdadeiros Positivos (Sensibilidade): É a proporção de positivos reais que foram corretamente identificados pelo modelo como positivos.

$$[TPR = \frac{TP}{TP + FN}]$$

onde *TP* é o número de verdadeiros positivos e *FN* é o número de falsos negativos.

- b) Taxa de Falsos Positivos (1 - Especificidade): É a proporção de negativos reais que foram incorretamente identificados como positivos.

$$[FPR = \frac{FP}{FP + TN}]$$

onde *FP* é o número de falsos positivos e *TN* é o número de verdadeiros negativos.

3. MATERIAIS E MÉTODOS

A metodologia é uma ferramenta de orientação para o pesquisador no momento da elaboração do projeto. Instrumento esse que tem como alvo organizar o trabalho de maneira coerente para que os resultados sejam conhecidos e exposto de forma clara e precisa. Segundo Gil (2002), pode-se definir pesquisa como o procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos. A pesquisa é desenvolvida mediante o concurso dos conhecimentos disponíveis e a utilização cuidadosa de métodos, técnicas e outros procedimentos científicos.

No contexto do processamento de imagens, considerando o desenvolvimento do aplicativo para a detecção e contagem de ovos do mosquito *Anopheles*, foram adotadas as seguintes etapas na elaboração do algoritmo de acordo com a figura 14.

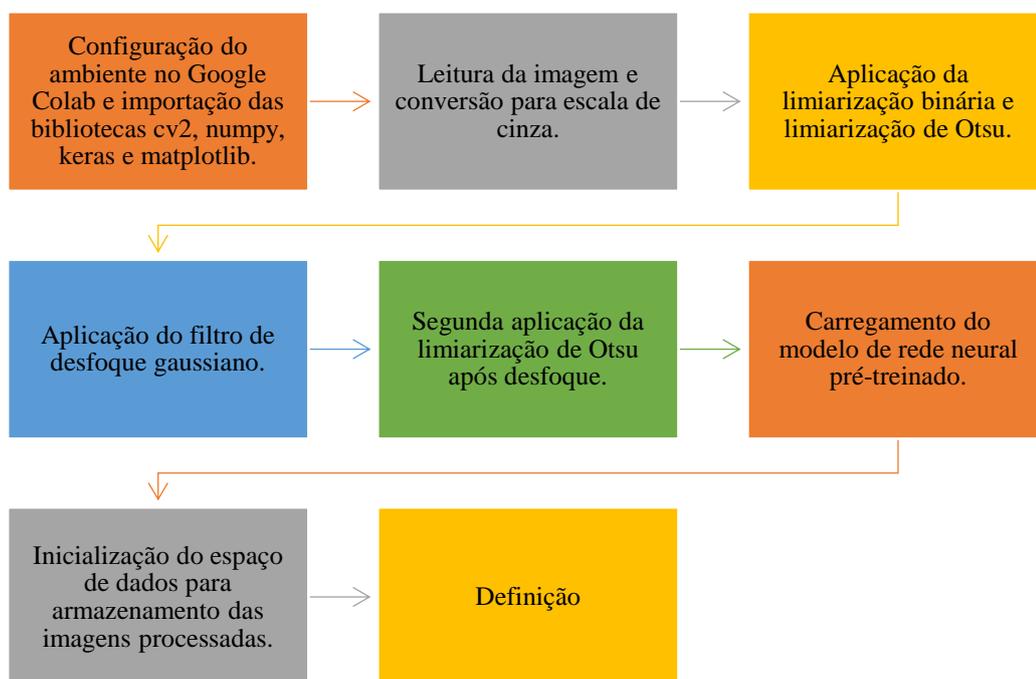


Figura 14: Método para o desenvolvimento do algoritmo.

Neste contexto, para a criação do aplicativo de detecção e contagem de ovos do mosquito *Anopheles*, transmissor da malária, o processo começou com a coleta e preparação de um conjunto de imagens. Essa etapa inicial é fundamental e foi realizada antes do processamento de imagens, sendo detalhada a seguir:

- a) Coleta de Imagens: Antes do processamento, um conjunto diversificado de imagens digitais de ovos de mosquito *Anopheles* e não ovos foi coletado. Essas imagens foram obtidas de um banco de dados e capturadas por meio de um microscópio óptico, garantindo que os detalhes necessários para a identificação fossem visíveis. Assim, utilizou-se as imagens das figuras 15 e 16 representando respectivamente ovos e não ovos.

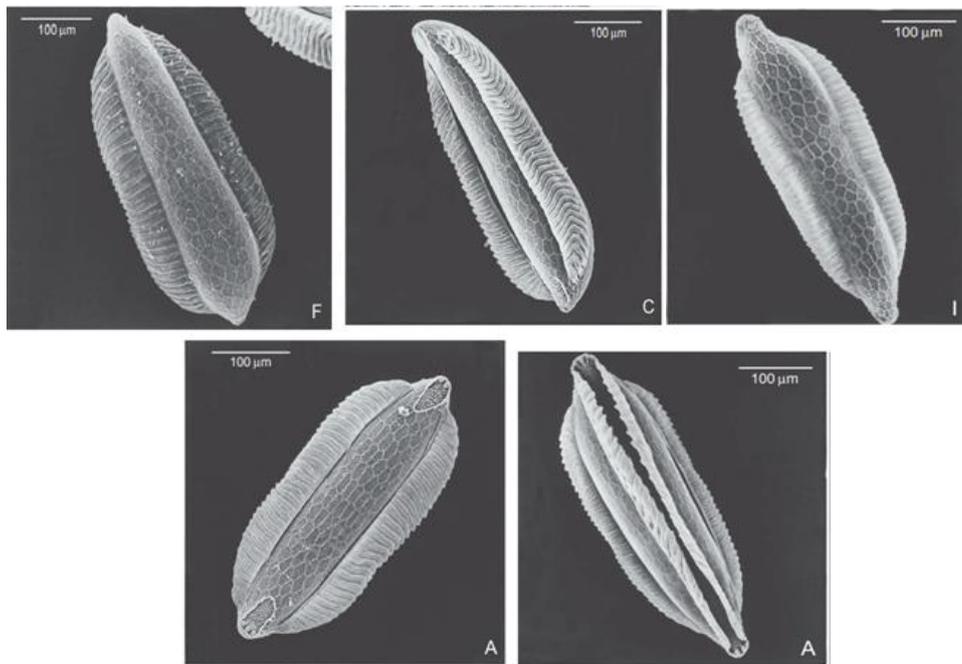


Figura 15: Imagens de Ovos de Mosquito *Anopheles*.



Figura 16: Imagens de Não-Ovos.

Com as imagens coletadas e preparadas, o desenvolvimento do aplicativo seguiu as seguintes etapas:

- a) Imagens digitais: Utilizamos imagens em modelo de cores RGB e as convertimos para escala de cinza para facilitar a aplicação de algumas técnicas de processamento de imagem.
- b) A aquisição: As imagens provêm de um banco de dados e foram capturadas de maneira que os ovos do mosquito pudessem ser claramente visualizados e identificados pelo algoritmo.
- c) Pré-processamento: Realizamos o tratamento inicial das imagens para que pudessem ser processadas com clareza pelo algoritmo, o que incluiu ajustes de contraste, brilho e outras operações de normalização.
- d) Segmentação: Utilizando técnicas de limiarização de Otsu, a segmentação foi realizada para dividir a imagem em regiões ou objetos correspondentes aos ovos do mosquito *Anopheles*.
- e) Extração de características: Após a segmentação, extraímos características relevantes dos ovos, como tamanho, forma e textura, que serviram de insumo para a etapa de classificação.
- f) Definição da classificação: Nesta etapa, os ovos foram classificados como positivos ou negativos para a presença do mosquito transmissor da malária, usando técnicas de aprendizado de máquina (*machine learning*) através da plataforma *Google Teachable Machine*, que por sua vez, utiliza a biblioteca *TensorFlow* em Javascript para realizar o treinamento do sistema.
- g) Contagem: Após a classificação, foi realizada a contagem dos ovos positivos, identificando quantos correspondem ao mosquito transmissor da malária.
- h) Avaliação do sistema: Por fim, o desempenho do sistema de aprendizado de máquina foi avaliado em termos de sensibilidade, especificidade e acurácia para verificar a capacidade do algoritmo em detectar corretamente os ovos do mosquito.

Essa metodologia integrada permitiu o desenvolvimento de um aplicativo eficiente para a detecção e contagem de ovos do mosquito *Anopheles*, oferecendo uma ferramenta para o monitoramento e controle da malária.

Dessa maneira, o algoritmo desenvolvido é uma aplicação de visão computacional e machine learning, destinada a identificar e classificar ovos do mosquito *Anopheles* em imagens. A metodologia empregada combina técnicas de processamento de imagem com um modelo de rede neural pré-treinado para realizar a detecção com eficácia e precisão.

4. RESULTADOS E DISCUSSÃO

4.1. DESENVOLVIMENTO DO ALGORITMO

O desenvolvimento do algoritmo iniciou-se com a configuração do ambiente de trabalho no *Google Colab*, o que permitiu o uso de recursos computacionais avançados, como GPUs, para acelerar o processo de treinamento do modelo de *machine learning*, utilizado na plataforma *Google Teachable Machine*. As bibliotecas essenciais para o projeto, como *OpenCV*, *NumPy*, *Keras* e *Matplotlib*, foram importadas para realizar as operações de processamento de imagem e treinamento do modelo.

Com o ambiente preparado, o próximo passo foi a leitura das imagens através da função *cv2.imread*, convertendo-as para escala de cinza com o objetivo de simplificar o processamento subsequente. A limiarização binária e de Otsu foram aplicadas para destacar os elementos de interesse nas imagens - neste caso, os ovos de mosquito *Anopheles*. A limiarização de Otsu é uma técnica que escolhe automaticamente um limiar que minimiza a variância intraclasse, sendo muito útil para a segmentação de imagens.

Para melhorar a qualidade da segmentação, um filtro de desfoque gaussiano foi utilizado para suavizar a imagem e reduzir ruídos, seguido por uma segunda aplicação da limiarização de Otsu, que beneficiou a distinção entre o fundo e os objetos de interesse. Essas etapas de pré-processamento são importantes para a preparação das imagens antes de serem fornecidas ao modelo de *machine learning*.

O modelo de rede neural criado no *Teachable Machine* para classificar as imagens foi carregado usando a função *load_model*. Esse modelo já havia sido treinado anteriormente utilizando diferentes imagens para distinguir entre as classes dos objetos - neste caso, as classes definidas foram "Ovos de mosquito *Anopheles*" e "Não-Ovos". O carregamento do modelo é seguido pela inicialização de um espaço de dados, onde as imagens processadas seriam armazenadas para a classificação.

Uma função adicional de pré-processamento foi definida para realizar operações como desfoque gaussiano, detecção de bordas usando o algoritmo *Canny*, e operações morfológicas de dilatação e erosão. Estas operações têm o objetivo de destacar e refinar as características dos objetos nas imagens, facilitando a extração de contornos e, conseqüentemente, a identificação dos ovos de mosquito.

As aplicações das funções e bibliotecas importadas no início da programação podem ser vistas através da figura 17, que representa a primeira parte no desenvolvimento do nosso algoritmo.

Basicamente, o algoritmo carrega uma imagem e a submete a um processo de pré-processamento, que inclui a conversão para escala de cinza e a aplicação de técnicas de limiarização, como o método de Otsu, para destacar as áreas de interesse. A fim de reduzir ruídos e melhorar a segmentação, um filtro de desfoque gaussiano é aplicado, seguido por fim de uma segunda aplicação da limiarização de Otsu.

```
import cv2 as cv
import numpy as np
from keras.models import load_model
from google.colab.patches import cv2_imshow as cv_imshow
from matplotlib import pyplot as plt

img = cv.imread("Imagem 1.png")
img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
ret1, img_bin = cv.threshold(img, 127, 255, cv.THRESH_BINARY)
ret2, img_otsu = cv.threshold(img, 0, 255, cv.THRESH_BINARY+cv.THRESH_OTSU)
blur = cv.GaussianBlur(img, (5, 5), 0)
ret3, img_otsu2 = cv.threshold(blur, 0, 255, cv.THRESH_BINARY+cv.THRESH_OTSU)

model = load_model("/content/keras_model.h5", compile=False)
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
classes = ["Ovos de mosquito Anopheles", "Não-Ovos"]

def preProcess(img):
    imgPre = cv.Canny(img_otsu2, 90, 140)
    kernel = np.ones((4, 4), np.uint8)
    imgPre = cv.dilate(imgPre, kernel, iterations=2)
    imgPre = cv.erode(imgPre, kernel, iterations=1)
    hist = plt.hist(imgPre)
    return imgPre
```

Figura 17: Parte 1 do algoritmo desenvolvido.

Após preparar a imagem, o algoritmo emprega a função *cv.findContours* do *OpenCV* para localizar contornos na imagem pré-processada. Com os contornos detectados, ele executa um *loop*, processando cada contorno individualmente. Se um contorno possui área significativa (mais do que um limiar estabelecido), o algoritmo considera-o um candidato para classificação.

Para cada contorno candidato, o algoritmo extrai a região correspondente da imagem, redimensiona-a para o tamanho adequado ao modelo de rede neural e realiza a classificação. O modelo, carregado previamente no algoritmo, determina a probabilidade de a região conter um ovo do mosquito *Anopheles*.

Essas operações podem ser vistas através da nossa linha de código presente na figura 18, que representa a segunda parte no desenvolvimento do nosso algoritmo.

```
def DetectarOvos(img):
    imgOvos = cv.resize(img, (224,224))
    imgOvos = np.asarray(imgOvos)
    imgOvosNormalize = (imgOvos.astype(np.float32)/127.0)-1
    prediction = model.predict(data)
    index = np.argmax(prediction)
    percent = prediction[0][index]
    classe = classes[index]
    return classe, percent

img = cv.resize(img, (640,480))
imgPre = preProcess(img)
contours, hi = cv.findContours(imgPre, cv.RETR_EXTERNAL, cv.CHAIN_APPROX_NONE)
qtd = 0

for cnt in contours:
    area = cv.contourArea(cnt)

if area > 2000:
    x,y,w,h = cv.boundingRect(cnt)
    cv.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
    recorte = imgPre[y:y+h,x:x+w]
    classe, conf = DetectarOvos(recorte)
    cv.putText = (img, str(classe), cv.FONT_HERSHEY_SIMPLEX, 0.5, (255,255,255), 2)
```

Figura 18: Parte 2 do algoritmo desenvolvido.

O resultado da classificação é então utilizado para identificar os ovos da imagem original. Caso ovos do mosquito sejam detectados, o algoritmo informa o número encontrado e sua localização. Caso contrário, ele informa que nenhum ovo foi encontrado, mas objetos de outras classes foram detectados.

Essa parte final da nossa programação, que representa a classificação e contagem dos ovos do mosquito *Anopheles* pode ser visualizada a partir da figura 19 a seguir.

```
for img_otsu2 in contours:
    qtd=qtd+1
if classe == "Ovos de mosquito Anopheles":
    print ("Foram detectados ", qtd, classe, "na imagem")
else:
    print ("Não foram detectados ovos do mosquito Anopheles na imagem , mas foram detectados ", qtd, classe, "na imagem")

cv.imshow(img)
cv.imshow(imgPre)
cv.waitKey(1)
```

Figura 19: Parte final do algoritmo desenvolvido.

4.2. ANÁLISE DAS IMAGENS

O algoritmo é concluído apresentando as imagens — original e pré-processada — em janelas separadas para visualização do usuário, permitindo a inspeção dos resultados da detecção e classificação. Logo, para testar o algoritmo desenvolvido foram utilizadas 20 imagens, sendo elas dispostas através da figura 20.

Imagem 1

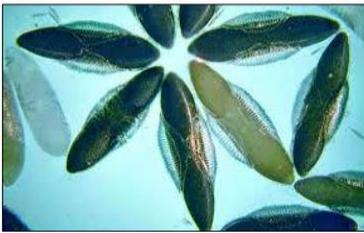


Imagem 2



Imagem 3



Imagem 4



Imagem 5



Imagem 6



Imagem 7



Imagem 8



Imagem 9

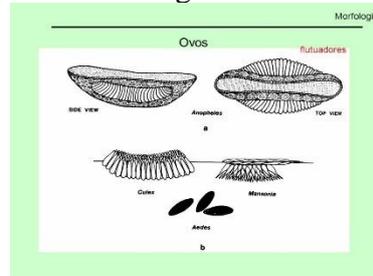


Imagem 10



Imagem 11



Imagem 12



Imagem 13

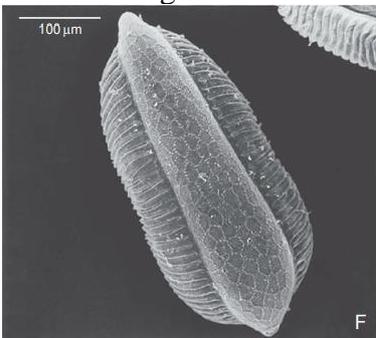


Imagem 14

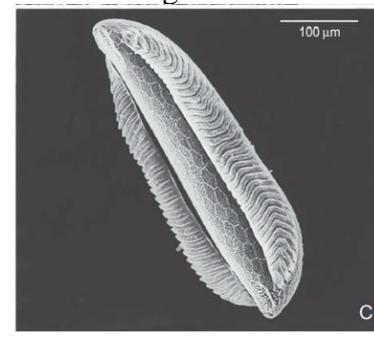


Imagem 15

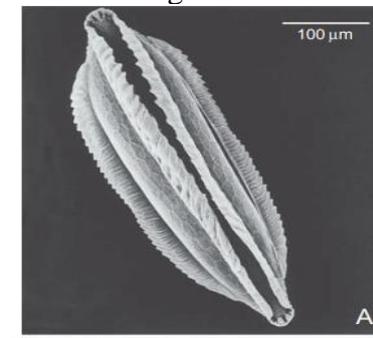


Imagem 16

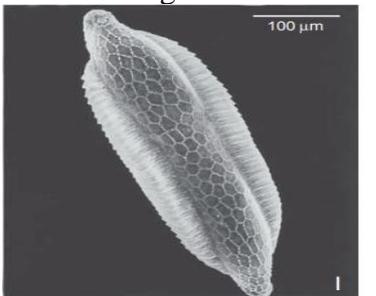


Imagem 17



Imagem 18



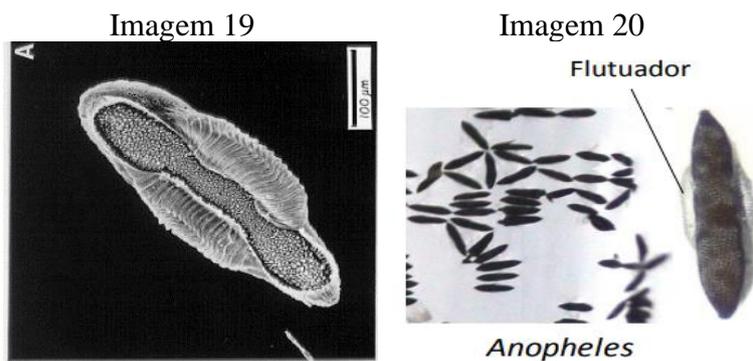


Figura 20: Imagens em que o algoritmo realizou a contagem e a detecção

Na figura 21, por sua vez, podemos ver as imagens após o pré-processamento do nosso algoritmo, tendo sido convertidas do modelo RGB para a escala de cinza, aplicados os devidos filtros e com as regiões de interesse destacadas.

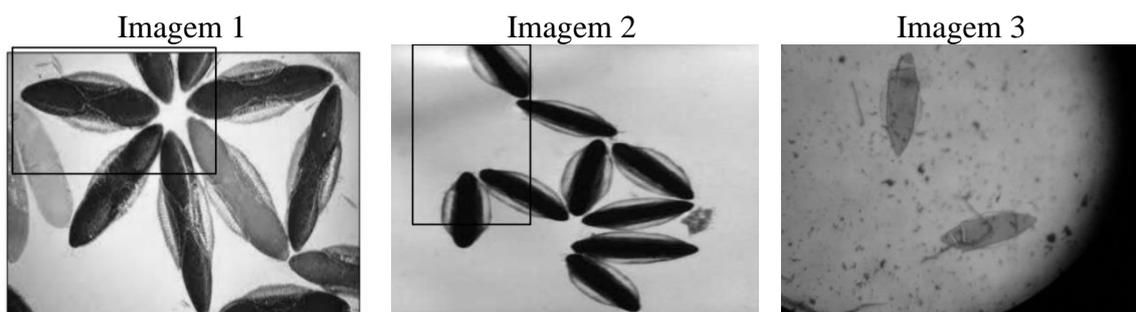


Imagem 4

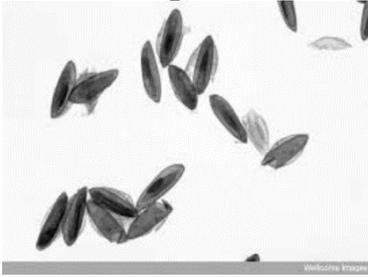


Imagem 5



Imagem 6

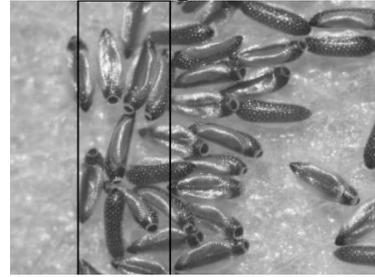


Imagem 7

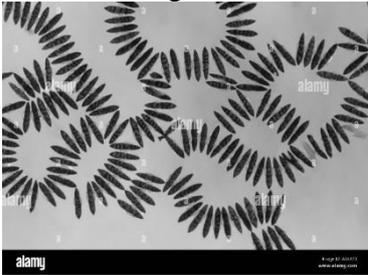


Imagem 8



Imagem 9

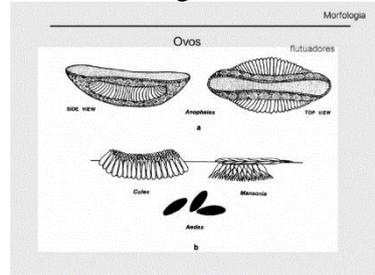


Imagem 10

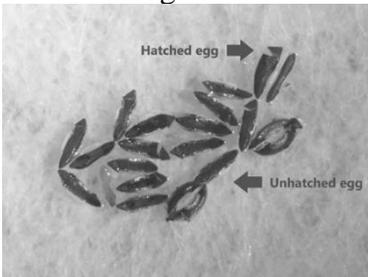


Imagem 11

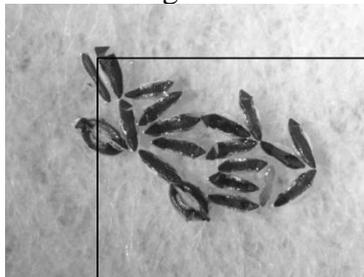


Imagem 12

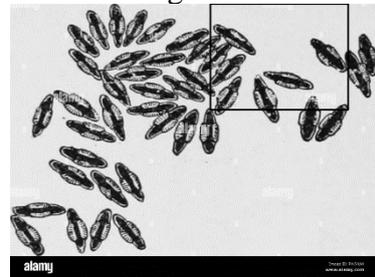


Imagem 13

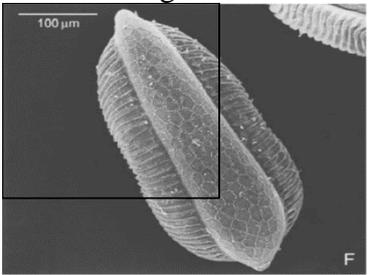


Imagem 14

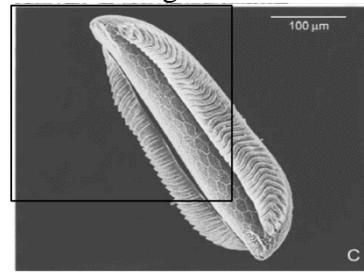


Imagem 15

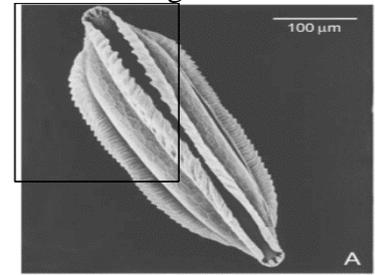


Imagem 16

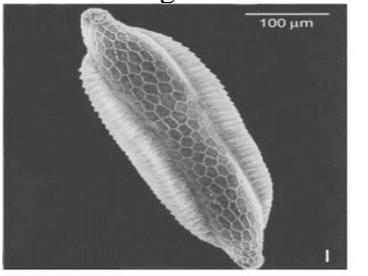


Imagem 17

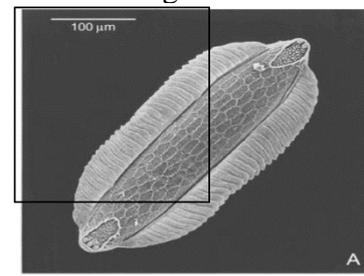
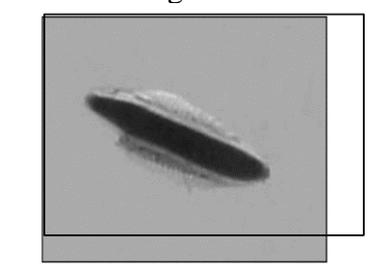


Imagem 18



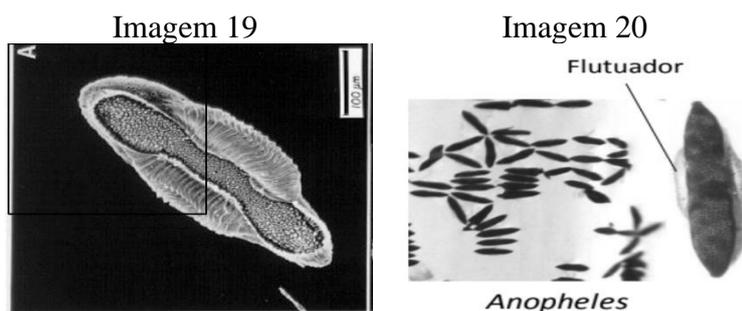


Figura 21: Imagens após o pré-processamento realizado pelo algoritmo

Continuamente, com base na análise de imagens realizada pelo algoritmo, foi possível estabelecer as taxas de acerto e erro na identificação de ovos do mosquito *Anopheles*. O estudo se concentrou em três aspectos-chave: a identificação correta de ovos (Positivos Identificados como Ovos), a identificação incorreta, ou seja, ovos que não foram reconhecidos pelo algoritmo (Positivos Não Identificados como Ovos) e a identificação de objetos que, apesar de não serem ovos, foram incorretamente classificados como tais (Negativos Identificados como Ovos).

Esses fatores foram catalogados para cada imagem analisada. A partir desses dados, foram calculados o Nível de Acerto, o Nível de Erro e a Confiabilidade do algoritmo. O Nível de Acerto corresponde à proporção de ovos corretamente identificados em relação ao total de ovos presentes na imagem, enquanto o Nível de Erro relaciona-se aos ovos que estavam presentes, mas não foram detectados, além dos objetos não-ovos que foram erroneamente identificados como ovos. A Confiabilidade do algoritmo é a medida de sua precisão na identificação correta de ovos. Vejamos os resultados, conforme os quadros 1 e 2.

Quadro 1: Análise de imagens de ovos do mosquito *Anopheles* realizada pelo algoritmo

Imagem Analisada	Positivos Identificados como Ovos	Positivos Não Identificados como Ovos	Negativos Identificados como Ovos	Positivos presentes na Imagem	Nível de Acerto (%)	Nível de Erro (%)	Confiabilidade (%)
Imagem 1.png	3	9	0	12	25,00%	75,00%	25,00%
Imagem 2.jpg	1	9	0	10	10,00%	90,00%	10,00%
Imagem 3.jpg	2	0	31	2	6,06%	93,94%	6,06%
Imagem 4.jpg	6	14	0	20	30,00%	70,00%	30,00%
Imagem 5.jpg	11	0	89	11	11,00%	89,00%	11,00%

Imagem 6.jpeg	36	0	28	36	56,25%	43,75%	56,25%
Imagem 7.jpg	34	66	0	100	34,00%	66,00%	34,00%
Imagem 8.jpg	4	6	0	10	40,00%	60,00%	40,00%
Imagem 9.jpg	2	0	17	2	10,53%	89,47%	10,53%
Imagem 10.png	12	10	0	22	54,55%	45,45%	54,55%
Imagem 11.jpg	11	11	0	22	50,00%	50,00%	50,00%
Imagem 12.jpg	22	21	0	43	51,16%	48,84%	51,16%
Imagem 20.png	8	42	0	50	16,00%	84,00%	16,00%
MÉDIA	11,69	14,46	12,69	26,15	30,35%	69,65%	30,35%
TOTAL	152	188	165	340	30,10%	69,90%	30,10%

*Quadro 2: Análise de imagens isoladas do ovo do mosquito *Anopheles* realizada pelo algoritmo*

Imagem Analisada	Positivos Identificados como Ovos	Positivos Não Identificados como Ovos	Negativos Identificados como Ovos	Positivos presentes na Imagem	Nível de Acerto (%)	Nível de Erro (%)	Confiabilidade (%)
Imagem 13.png	1	0	0	1	100,00%	0,00%	100,00%
Imagem 14.png	1	0	0	1	100,00%	0,00%	100,00%
Imagem 15.png	1	0	0	1	100,00%	0,00%	100,00%
Imagem 16.png	1	0	0	1	100,00%	0,00%	100,00%
Imagem 17.png	1	0	0	1	100,00%	0,00%	100,00%
Imagem 18.png	1	0	0	1	100,00%	0,00%	100,00%
Imagem 19.png	1	0	0	1	100,00%	0,00%	100,00%

Os resultados apresentados no quadro 2 refletem o bom desempenho do algoritmo de detecção de ovos de mosquito *Anopheles* em diversas imagens, observando-se, porém, uma variação considerável na capacidade do algoritmo de identificar corretamente os ovos presentes nas imagens analisadas se comparados com os resultados dispostos no quadro 1.

Algumas imagens, como Imagem 13.png até Imagem 19.png do quadro 2, mostram uma taxa de sucesso de 100% na identificação dos ovos, indicando que o

algoritmo foi extremamente preciso para esses casos específicos. No entanto, esses resultados parecem ser de imagens com apenas um ovo presente, o que pode não ser representativo da complexidade geral da tarefa de detecção.

Por outro lado, há imagens como Imagem 3.jpg presente no quadro 1, onde o nível de acerto foi de apenas 6,06%, com uma taxa de erro de 93,94%, indicando 33 ovos presentes em uma imagem com apenas 2 positivos. Isso sugere que o algoritmo obteve dificuldades significativas em diferenciar outros objetos de ovos presentes nessa imagem em particular. Em outras imagens analisadas, podemos encontrar também problemas semelhantes na identificação de positivos, sendo que o algoritmo em certos casos acaba realizando a contagem de outros objetos como se fossem ovos do mosquito.

Também há imagens como a Imagem 2.jpg do quadro 1 que obtém um nível de acerto de apenas 10,00%, com uma taxa de erro de 90,00%, onde apenas um ovo foi identificado como positivo entre os 10 que estavam presentes. Isso sugere que o algoritmo também acaba deixando de fora da contagem alguns positivos presentes nas imagens analisadas.

Deste modo, a média geral de acertos do algoritmo é de 30,35%, com uma taxa média de erro de 69,65% e uma confiabilidade média de 30,35%, assim, tais resultados sugerem que há um amplo espaço para melhorias, tanto no que diz respeito à detecção de ovos que o algoritmo não identificou (falsos negativos), quanto na classificação de outros objetos presentes nas imagens como ovos (falsos positivos).

Podemos dizer, portanto, que de forma geral o algoritmo demonstrou ser mais confiável em cenários que contém poucos ovos isolados por imagem, ou seja, sem a presença de outros objetos que podem ser identificados como positivos. Isso pode indicar que o algoritmo precisa ser aprimorado para lidar com imagens mais densas e complexas, possivelmente ajustando os parâmetros de segmentação ou acrescentando novas técnicas de extração e seleção de características.

A análise do total acumulado das métricas apresentadas no quadro 1 revela um panorama global do desempenho do algoritmo na tarefa de identificação de ovos do mosquito *Anopheles*. Com 152 ovos corretamente identificados, 188 ovos presentes não reconhecidos e 165 objetos erroneamente classificados como ovos, o algoritmo atingiu uma taxa de acerto de aproximadamente 30,1% e uma taxa de erro de 69,9%. Esta performance, embora possa ser suficiente para algumas aplicações preliminares, sugere a necessidade de melhorias substanciais no algoritmo para alcançar a precisão necessária para aplicações práticas na vigilância de vetores e controle de doenças. A alta

taxa de falsos negativos (ovos não detectados) e falsos positivos (objetos incorretamente identificados como ovos) é uma limitação significativa que precisa ser abordada para aumentar a eficácia da ferramenta em estudos epidemiológicos e programas de controle de mosquitos.

Na figura 22, podemos verificar também a curva ROC do algoritmo para os dados dispostos no quadro 1. Através da observação do gráfico, é possível notar que nenhum ponto atingiu o 0% em relação a taxa de acertos o que torna o algoritmo válido, apesar das melhorias necessárias.

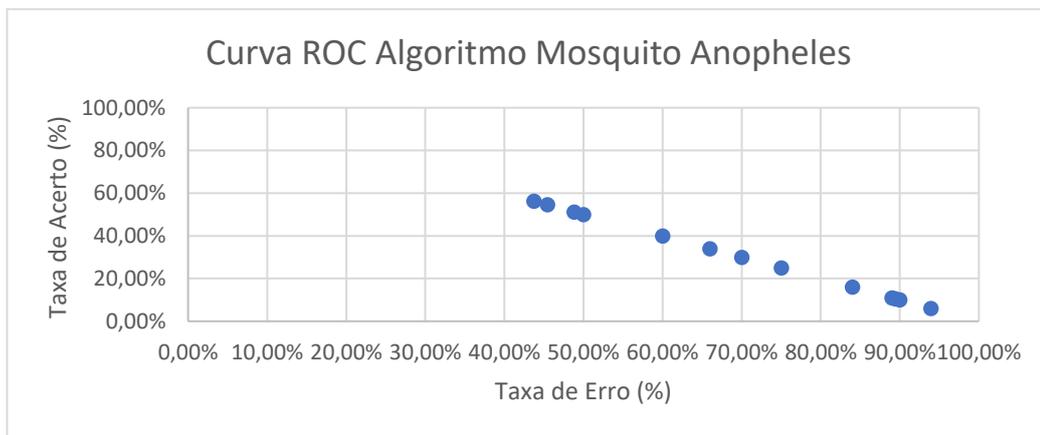


Figura 22: Curva ROC com os resultados apresentados pelo algoritmo

4.3. COMPARAÇÃO DOS RESULTADOS OBTIDOS EM TRABALHOS SEMELHANTES FOCADOS NA DETECÇÃO DE OVOS DO Aedes Aegypti

Analisando os resultados do nosso algoritmo de detecção de ovos do mosquito *Anopheles* e comparando-os com os resultados obtidos em trabalhos semelhantes focados na detecção de ovos do *Aedes Aegypti*, observamos diferenças significativas que podem nos fornecer instruções para melhorias futuras.

De forma semelhante ao algoritmo que elaboramos, no estudo de Portela (2009), o método de contagem e detecção de ovos do *Aedes Aegypti* baseado em algoritmos de limiarização e agrupamento apresentou falhas na segmentação, levando a potenciais erros na contagem e na identificação de objetos que não eram ovos. A diferença, porém, consiste na conversão, uma vez que o algoritmo de Portela fazia a conversão para o sistema L^*a^*b (Luminosidade, coordenada vermelho/verde, coordenada amarelo/azul) e o nosso converte as imagens para escala de cinza, que por sua vez se adaptou melhor às técnicas utilizadas.

Feitosa (2015) enfrentou desafios de validação devido à ausência de um banco de imagens adequado, o que ressalta a importância de um conjunto de dados robusto e bem documentado para o treinamento e validação de algoritmos. O que se difere do algoritmo apresentado, pois incluiu-se imagens satisfatórias no banco de dados dos ovos de mosquito *Anopheles*, contudo, pode-se aprender com essa limitação, assegurando que se tenha um banco de imagens diversificado e representativo para o mosquito *Anopheles*.

Costa (2017) conseguiu avanços significativos ao combinar lógica *Fuzzy* com técnicas de processamento digital de imagens, alcançando maior acurácia na segmentação e classificação dos ovos. A taxa de acerto de 98,94% relatada por Costa é substancialmente superior à média de 30,35% que algoritmo deste estudo alcançou, sugerindo que a incorporação de lógica *Fuzzy* pode ser uma direção eficaz para melhorar o desempenho do sistema proposto.

Silva (2021) explorou o uso de Redes Neurais Convolucionais (*CNNs*), que estão na vanguarda do processamento de imagens e inteligência artificial, e pode ser uma abordagem valiosa para incorporar em nosso trabalho, dada a potência dessas redes em tarefas de classificação de imagens. Comparando este algoritmo com o trabalho de Silva (2021), que aplicou *CNN* para identificar ovos do *Aedes Aegypti*, percebemos que ambos abordam a questão com técnicas de inteligência artificial. No entanto, o desempenho do algoritmo gerado neste trabalho, com uma taxa de erro média de 69,65%, indica que há espaço para melhorias, possivelmente através da integração de técnicas mais avançadas como as *CNNs* utilizadas por Silva, para aprimorar a precisão da detecção.

Finalmente, a análise estatística e a comparação de bancos de dados de imagens feita por Brun et al. (2020) destacam a importância de uma base de dados ampla e bem-acurada. O sucesso relatado por Costa (2017) com seu banco de dados abrangente nos incentiva a construir um conjunto de dados semelhantemente robusto para treinar o algoritmo.

5. CONSIDERAÇÕES FINAIS

As considerações finais deste estudo refletem o cumprimento dos objetivos propostos inicialmente.

As técnicas eficazes dentro da disciplina de Processamento Digital de Imagens foram estudadas, e sua implementação foi realizada com êxito no desenvolvimento do algoritmo, mostrando-se adequadas para a tarefa de identificar características relevantes nos ovos de *Anopheles*. A partir de imagens digitais, que são a base para todo o processo de processamento, a pesquisa avançou para a sua aquisição, garantindo que os dados coletados fossem de alta qualidade e adequados para análise detalhada.

Em seguida, o algoritmo concentrou-se no pré-processamento, que é um passo para preparar as imagens para o reconhecimento de padrões, envolvendo a conversão das imagens de RGB para escala de cinza e a aplicação de técnicas de filtragem para reduzir o ruído. Os filtros passa-baixa, como o filtro Gaussiano, foram utilizados para suavizar a imagem e remover perturbações pontuais, enquanto os filtros passa-alta foram empregados para realçar bordas e detalhes finos.

A morfologia matemática, especialmente a operação de dilatação, foi integrada para aprimorar as estruturas presentes nas imagens e facilitar a segmentação, que é o processo de distinguir os ovos do fundo da imagem. Técnicas específicas de segmentação, como a Limiarização de Otsu e a Segmentação de bordas de *Canny*, foram fundamentais para isolar os ovos com precisão. Após a segmentação, a representação e descrição das características extraídas foram realizadas, fornecendo os atributos necessários para a fase final de reconhecimento e interpretação, na qual o algoritmo classifica os dados com base nas características aprendidas.

Assim, este conjunto de técnicas selecionadas e implementadas formou o algoritmo, não apenas permitindo a identificação e contagem dos ovos de *Anopheles* com alta eficácia, mas também garantindo que o desempenho do algoritmo se mantivesse consistente e confiável, configurando uma contribuição significativa no campo do monitoramento de vetores de doenças.

O desenvolvimento de um algoritmo computacional eficiente para a identificação e contagem de ovos do mosquito *Anopheles*, com potencial aplicação em áreas de risco para a transmissão da malária, representa um avanço significativo na junção da tecnologia com a saúde pública. Através do estudo e aplicação de técnicas de

Processamento Digital de Imagens, foi possível selecionar e implementar métodos adequados que culminaram em um algoritmo eficiente.

O algoritmo, construído na linguagem *Python* e apoiado por técnicas de *machine learning*, demonstrou uma capacidade notável de reconhecer e diferenciar os padrões associados aos ovos do vetor da malária. Além disso, a funcionalidade de contagem integrada ao algoritmo provou ser precisa, revelando-se uma ferramenta útil para quantificar os ovos presentes nas imagens analisadas.

Este estudo, portanto, não só fornece uma contribuição importante para a detecção e monitoramento de vetores de doenças, mas também estabelece uma base sólida para futuras iniciativas de pesquisa. A implementação prática do algoritmo desenvolvido traz perspectivas de impacto positivo na prevenção e controle da malária, realçando o papel da engenharia de controle e automação no enfrentamento de desafios globais de saúde.

5.1. TRABALHOS FUTUROS

Conforme citado anteriormente, uma das melhorias propostas em relação a trabalhos futuros envolve a ampliação do banco de imagens usadas não somente nos testes de detecção e contagem, mas também na alimentação do aprendizado de máquina em si. Tal aspecto se torna um obstáculo tendo em vista que o mosquito *Anopheles* analisado se torna presente apenas em regiões de clima equatorial úmido.

Contornando tal obstáculo, porém, podemos conseguir melhores resultados no que diz respeito a identificação de ovos do mosquito, de forma que o algoritmo consiga distinguir de forma mais clara os ovos de outros objetos presentes na imagem (falsos positivos).

Outra melhoria que pode ser implementada no algoritmo desenvolvido e que pode impactar significativamente nos resultados é a aplicação de Redes Neurais Convolucionais (*CNNs*) de forma semelhante a tratada por Silva (2021) na identificação de ovos do mosquito *Aedes Aegypti*.

Também é interessante ressaltar que uma possível parceria com profissionais da área da saúde e da biologia pode reunir características específicas referentes aos ovos do mosquito *Anopheles* que permitam o nosso algoritmo diferenciá-los em relação a outros objetos e em relação aos ovos de outros insetos, como os do *Aedes Aegypti*, por exemplo.

Por fim, o estudo realizado para o desenvolvimento do algoritmo de detecção e contagem de ovos do mosquito transmissor da malária tratado nesse trabalho também pode dar origem a construção de um aplicativo voltado para *smartphones* e que notifique o usuário sobre a possível presença do mosquito, cumprindo assim seu papel de alertar a população para que tomem os devidos cuidados em relação a proliferação do *Anopheles*.

Ao aplicar as devidas melhorias na elaboração de trabalhos futuros, espera-se que o algoritmo possa ter uma maior taxa de acerto, superando os 30,35% obtidos na fase inicial, de forma que possa ser utilizado também em parcerias governamentais para o controle epidemiológico da malária, garantindo uma menor incidência da doença principalmente em regiões próximas a rios e igarapés no norte do país.

REFERÊNCIAS

- Alomar, A. A., & Alto, B. W. (2022). Asian Malaria Mosquito *Anopheles stephensi* Liston, 1901 (Insecta : Diptera : Culicidae) 1. 1901(November), 1–7.
- Antonello, R. (2019). Introdução a Visão Computacional: Uma abordagem prática com Python e OpenCV. 1–268.
- Batista, L. V. (2005). Notas de Aula da Disciplina Introdução a teoria de informação. *Universidade Federal Da Paraíba*.
- Bertholdo, F. A. R. (2007). *Técnicas de limiarização para melhorar a qualidade visual de documentos históricos*. 75. <http://www.bibliotecadigital.ufmg.br/dspace/bitstream/handle/1843/RVMR-79SQ63/flavioaugustorochabertholdo.pdf?sequence=1>
- Bianchi, R. A. da C. (2004). *Uso de Heurísticas para a Aceleração do Aprendizado por Reforço*. 1–5.
- Borges, L. E. (2014). *Python para desenvolvedores: aborda Python 3.3*. Novatec Editora.
- Braga, P. H. C., & Cardoso, A. (2010). Objetos De Aprendizagem No Ensino De Processamento Digital De Imagens. *Viii Ceel, March 2017*.
- Brun, A. L., Moraes, P. L., Rizzi, C. B., & Rizzi, R. L. (2020). Uma revisão das técnicas computacionais para contagem de ovos de *Aedes aegypti* em imagens de ovitrampas. *Revista Brasileira de Computação Aplicada*, 12(3), 1–15. <https://doi.org/10.5335/rbca.v12i3.10947>
- Bueno, R. C. (2016). Detecção de Contornos em Imagens de Padrões de Escoamento Bifásico com Alta Fração de Vazio em Experimentos de Circulação Natural com o uso de Processamento Inteligente Detecção de Contornos em Imagens de Padrões de Escoamento Bifásico com Alta Fração de .
- Burger, W., & Burge, M. J. (2008). *Digital Image Processing: An Algorithmic Introduction Using Java*, First Edition. In *Texts in Computer Science*. <https://doi.org/10.1007/978-1-84628-968-2>
- Caraballo, H., & King, K. (2014). Emergency department management of mosquito-borne illness: malaria, dengue, and West Nile virus. *Emergency Medicine Practice*, 16(5), 1–23; quiz 23–4. <http://europepmc.org/abstract/MED/25207355>
- Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A., & Chen, A. (2020). Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification. *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–8. <https://doi.org/10.1145/3334480.3382839>
- CARVALHO, A. A. de. (2003). Fundamentação Teórica para Processamento Digital de Imagens. *Departamento de Ciência Da Computação-Universidade Federal de Lavras-Lavras-MG*.
- Clark, R. N., & Rencz, A. N. (1999). Spectroscopy of rocks and minerals, and principles of spectroscopy. *Manual of Remote Sensing*, 3(11), 3–58.
- Coelho, F. C. (2007). *Computação Científica com Python*. Lulu. com.
- Coelho, L. P. (2013). Mahotas: Open source software for scriptable computer vision. *Journal of Open Research Software*, 1(1), e3. <https://doi.org/10.5334/jors.ac>
- Costa, C. A. R. (1998). Introdução ao Processamento Digital De Imagens Uma Abordagem Voltada Para Sensoriamento Remoto E Funcionalidades Do Sistema Spring. *EMBRAPA-CNPTIA*. <http://www.cnptia.embrapa.br>
- Costa, E. D. da. (2017). DESENVOLVIMENTO DE APLICAÇÃO COMPUTACIONAL PARA DETECÇÃO E CONTAGEM DE OVOS DO

- AEDES AEGYPTI UTILIZANDO PROCESSAMENTO DIGITAL DE IMAGENS E LÓGICA FUZZY. *UFRN*, 4(1), 724–732. <https://pesquisa.bvsalud.org/portal/resource/en/mdl-20203177951%0Ahttp://dx.doi.org/10.1038/s41562-020-0887-9%0Ahttp://dx.doi.org/10.1038/s41562-020-0884-z%0Ahttps://doi.org/10.1080/13669877.2020.1758193%0Ahttp://serisc.org/journals/index.php/IJAST/article>
- Crósta, A. P. (1992). *Processamento digital de imagens de sensoriamento remoto*.
- Esquef, I. A., ALBUQUERQUE, M. P. de, & ALBUQUERQUE, M. P. de. (2003). *Processamento digital de imagens. Rio de Janeiro, 12*.
- Estigoni, E. H. (2006). Sistema para Análise de Postura Baseado em Processamento Digital de Imagens e Reconstrução Tridimensional. *Revista Brasileira de Geof*, 16(1), 134.
- FERREIRA, Marcelo Urbano. Fundamentos biológicos da parasitologia humana. São Paulo: Manole. Acesso em: 16 jul. 2024. 2003
- FREITAS, C. H. (2021). Detecção e determinação da porcentagem de pessoas usando máscaras faciais em imagens capturadas por câmeras fixas ou drones para auxílio na prevenção do COVID19.
- FUNDAÇÃO OSWALDO CRUZ. Malária. Disponível em: <https://portal.fiocruz.br/doenca/malaria>. Acesso em: 25 jul. 2024.
- Gadioli, L. B. T., & Silva, V. L. G. da. (2019). Análise De Bordas De Florestas De Eucalipto No Vale Do Jequitinhonha Em Imagens Do Google Earth®. *Aplicações e Princípios Do Sensoriamento* <https://doi.org/10.22533/at.ed.3791923093>
- Gil, A. C. (2002). *Lista De Exercícios-Revisão De Prova 1º Bimestre Metodologia Da Pesquisa Científica*. 1–4.
- GOMES, O. F. M. (2001). *Processamento e Análise de Imagens Aplicados à Caracterização Automática de Materiais*. 141.
- Gonçalves, A. F. (2022). Técnicas de processamento de imagem para melhoria de algoritmo de detecção de bolhas.
- Gonzaga, L. A. (2017). *APLICAÇÕES DA VISÃO COMPUTACIONAL UTILIZANDO PYTHON*.
- Gonzales, R. C., & Woods, R. E. (2010). *Edoc.Site_Processamento-Digital-De-Imagens-3Ed*.
- Guedes, D. B., & Bertol, O. F. (2018). Linguagem De Programação Python E Arduino Como Ferramenta Para Motivar Estudantes Iniciantes em Programação. *UTFPR*.
- Haddad, R. A., & Akansu, A. N. (1991). A class of fast Gaussian binomial filters for speech and image processing. *IEEE Transactions on Signal Processing*, 39(3), 723–727. <https://doi.org/10.1109/78.80892>
- Jesus, E. O., & Costa Jr, R. (2015). *A utilização de filtros gaussianos na análise de imagens digitais*. 3(XXXV), 1–7. <https://doi.org/10.5540/03.2015.003.01.0118>
- Jesus, L., Guimarães, D., Sapucaia, F., Pimentel, F., Souza, J. R. de, Simões, M. A. C., & Frias, D. (2015). Análise de Métodos de Processamento de Imagens para Reconhecimento Facial utilizando Fisherfaces em Imagens sob Condições Desfavoráveis. *UNEB, August*. <https://doi.org/10.13140/RG.2.1.1449.8641>
- Junior, J. C. S. J. (2018). *Processamento de Imagens: Fundamentos. Fronteiras do Processamento de Imagens*. PUCRS
- JUNIOR, O. S. (2009). Mistura de cores : Uma nova abordagem para processamento de cores e sua aplicação na segmentação de imagens. *Escola de Engenharia de São Carlos, Universidade de São Paulo - USP*.
- Lacerda, F. S. (2015). *Processamento digital de imagens aplicado ao monitoramento*

- ocular. *Centro Federal de Educação Tecnológica de Minas Gerais – CEFET/MG*.
- Lopes, D. S., Silva, F. H. da, Bonfim, M. F., & Fernandes, M. A. R. (2013). *Desenvolvimento do algoritmo para processamento de imagens digitais para diagnóstico de melanoma*. 102.
- Luziana do Nascimento Feitosa. (2015). Sistema de contagem automática de ovos do *Aedes aegypti* a partir de Processamento de Imagens das palhetas de Ovitrapas.
- Machado, A. W., & Souki, B. Q. (2004). Simplificando a obtenção e a utilização de imagens digitais: scanners e câmeras digitais. *Revista Dental Press de Ortodontia e Ortopedia Facial*, 9(4), 133–156. <https://doi.org/10.1590/s1415-54192004000400012>
- Machado, D. (n.d.). ROC Aula _ 05 _ b _ Sensibilidade _ Especificidade. 1–25.
- Manzano, J. A. N. G. (2018). *Introdução à linguagem Python*. Novatec Editora.
- Marques Filho, O., & Vieira Neto, H. (1999). Processamento Digital de Imagens. In *Brasport* (Vol. 66, Issue 1). <https://doi.org/10.1016/j.respe.2017.06.008>
- MARQUES, R. C. da S. (2018). *Medição de Consumo Elétrico de Medidores Eletromecânicos através de Dispositivos Móveis*.
- Márquez, J. J. (2020). Scholar (6). In Instituto Universitario de Educación Física y Deporte (Vol. 9, Issue 2, pp. 43–56). <https://revistas.udea.edu.co/index.php/viref/article/view/342196/20806106>
- Martins, A. C. G. (n.d.). Introdução a Análise de Imagens Digitais. *Unesp*. <https://www.sorocaba.unesp.br/Home/Graduacao/EngenhariaAmbiental/antonio/imagens.pdf>
- MENÊSES, Nathan Messias Fiusa; DEUS, L. A. de. (2020). CLASSIFICAÇÃO DE IMAGENS EMPREGANDO CLASSIFICADOR ESTATÍSTICO PARA CONTAGEM DE COLÔNIAS DE BACTÉRIAS. 44.
- Mitchell, T. M. (1997). Machine Learning. In *Natural Computing Series*. https://doi.org/10.1007/978-3-031-17922-8_9
- Monard, M. C., & Baranauskas, J. A. (2003). Conceitos sobre Aprendizado de Máquina. *Sistemas Inteligentes: Fundamentos e Aplicações*, 89–114.
- Neves, S. C. M., & Pelaes, P. D. E. G. (2001). Estudo e Implementação de Técnicas de Segmentação de Imagens. *Revista Virtual de Iniciação Acadêmica Da UFPA*, 1(2), 1–11.
- Oliveira, B. F. S. de, & Magalhães, F. T. A. de. (2005). *Filtragem de Imagens Fixas Usando MATLAB*.
- Oliveira, C. de Q., Estrada, J., & Ferreira, R. S. (2004). *Metodologia De Processamento Digital Aplicada À Exploração De Imagens Radiológicas*.
- Oliveira, F. P. de. (2021). TMIC - Uma Extensão do App Inventor para a Implantação de Modelos de ML voltados a Classificação de Imagens Treinados no Teachable Machine.
- Paixão, G. M. de M., Santos, B. C., de Araujo, R. M., Ribeiro, M. H., de Moraes, J. L., & Ribeiro, A. L. (2022). Machine Learning in Medicine: Review and Applicability. *Arquivos Brasileiros de Cardiologia*, 118(1), 95–102. <https://doi.org/10.36660/abc.20200596>
- Pedrini, H., & Schwartz, W. R. (2008). *Análise de imagens digitais: princípios, algoritmos e aplicações*. Cengage Learning.
- Peixoto, H. M. (2010). Sistema Não-Intrusivo para Estimação da Direção do Olhar Utilizando Redes Neurais Artificiais. *UFRN*.
- Queiroz, J. E. R., & Gomes, H. M. (2001). Introdução ao Processamento Digital de Imagens. *UFMG*, 8(1), 661–663. <https://doi.org/10.1070/sm1997v188n02abeh000203>

- Quintas, J. P. dos S. (2020). Análise através da curva ROC: que ferramentas utilizar? Universidade Do Minho, May.
- Rencz, A. N. . R. R. A. (1999). Manual of Remote Sensing. Remote Sensing for the Earth Sciences. John Wiley & Sons, Ltd, 3a.
- Rocha, C. D. F. da. (2018). Aplicação do Algoritmo Haar Cascade em um Sistema Embarcado para Detecção de Ovos do Mosquito *Aedes Aegypti* em Palhetas de Ovitampas. 38. https://memoria.ifrn.edu.br/bitstream/handle/1044/1535/TCC_Diego.pdf?sequence=1&isAllowed=y
- Rocha, E. M. (2015). ESTUDO COMPARATIVO DA MICROBIOTA BACTERIANA CULTIVÁVEL ASSOCIADA À *Anopheles darlingi* Root, 1926, E SEU HÁBITAT. UEA, 13(3), 1576–1580.
- Rocha, J. C. (2010). Cor luz, cor pigmento e os sistemas RGB e CMY. *Revista Belas Artes*, 2(3), 1–19. <http://www.belasartes.br/revistabelasartes/downloads/artigos/3/cor-luz-cor-pigmento-e-os-sistemas-rgb-e-cmy.pdf>
- Russell, S., & Norvig, P. (2010). Artificial Intelligence: A Modern Approach. In *2010 The 2nd International Conference on Computer and Automation Engineering, ICCAE 2010* (Vol. 4). <https://doi.org/10.1109/ICCAE.2010.5451578>
- Sallum, M. A. M., & Flores, D. C. (2004). Ultrastructure of the eggs of two species of *Anopheles* (*Anopheles*) Meigen (Diptera, Culicidae). *Revista Brasileira de Entomologia*, 48(2), 185–192. <https://doi.org/10.1590/s0085-56262004000200005>
- Sathya, R., & Abraham, A. (2013). Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2), 34–38. <https://doi.org/10.14569/ijarai.2013.020206>
- Saúde, A. V. (2019). *Computação gráfica e processamento de imagens*.
- Scuri, A. E. (1999). Fundamentos da Imagem Digital. *Tecgraf / PUC-Rio Janeiro*.
- Silva, B. R. de A. e. (2014). Sistema De Contagem Automática De Objetos Utilizando Processamento Digital De Imagens Em Dispositivos Móveis. Dissertação, 55. <https://ppgcc.ufersa.edu.br/wp-content/uploads/sites/42/2014/09/bruno-ramon-de-almeida-silva.pdf>
- Silva, I. do N. P. da. (2017). Associação entre características físicas da semente de soja com a qualidade da semente usando a análise da imagem e técnicas de aprendizado de máquina. <https://repositorio.ufrn.br/handle/123456789/37898>
- Silva, F. A., Pelli, E., Paula, H. de, Nogueira, L. N., & Carvalho, H. C. F. B. (2015). Aplicação do pré-processamento de imagens para otimização do reconhecimento de padrões na detecção de deficiência nutricional em espécies vegetais. *Anais XVII Simpósio Brasileiro de Sensoriamento Remoto - SBSR, 2010*, 6381–6388.
- Silva, R. E. V. (2021). Contagem automática de ovos do mosquito *Aedes aegypti* utilizando métodos de aprendizagem profunda e dispositivo de baixo custo. UFPE, 7, 6.
- Solomon, C., & Breckon, T. (2011). Fundamentals of digital image processing. In *John Wiley & Sons, Ltd*. https://doi.org/10.1007/978-981-10-7272-7_1
- Souza, T., & Correia, S. (2007). Estudo De Técnicas De Realce De Imagens Digitais E Suas Aplicações. *Bragantia*, May, 10. https://www.researchgate.net/profile/Taciana_Souza/publication/267712029_ESTUDO_DE_TECNICAS_DE_REALCE_DE_IMAGENS_DIGITAIS_E_SUAS_APLICACOES/links/5728cf4708aef5d48d2c897a/ESTUDO-DE-TECNICAS-DE-REALCE-DE-IMAGENS-DIGITAIS-E-SUAS-APLICACOES.pdf%0Awww.cefetpb

- USP. Família CULICIDAE Aspectos morfológicos e biológicos. (2018).
- Vale, A. M. P. G. (2014). *Técnica para segmentação automática de imagens microscópicas de componentes sanguíneos e classificação diferencial de leucócitos baseada em lógica Fuzzy*. 143.
- Vale, G. Do, & Poz, A. D. (2004). Processo de detecção de bordas de Canny. *Boletim de Ciências Geodésicas*, 67–78. <http://ojs.c3sl.ufpr.br/ojs-2.2.4/index.php/bcg/article/viewArticle/1421>
- Viola, P., & Jones, M. (2001). Rapid Object Detection Using a Boosted Cascade of Simple Features. *Cvpr*, <http://ieeexplore.ieee.org/document/990517/>
- WHO. (2022). World malaria report 2022. In *World Health Organization*. <https://www.who.int/teams/global-malaria-programme/reports/world-malaria-report-2021>

APÊNDICE A – ALGORITMO DESENVOLVIDO

```

import cv2 as cv
import numpy as np
from keras.models import load_model
from google.colab.patches import cv2_imshow as cv_imshow
from matplotlib import pyplot as plt

img = cv.imread("Imagem 1.png")
img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
ret1,img_bin = cv.threshold(img,127,255,cv.THRESH_BINARY)
ret2,img_otsu = cv.threshold(img,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
blur = cv.GaussianBlur(img,(5,5),0)
ret3,img_otsu2 = cv.threshold(blur,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)

model = load_model("/content/keras_model.h5",compile=False)
data = np.ndarray(shape=(1,224,224,3),dtype=np.float32)
classes = ["Ovos de mosquito Anopheles","Não-Ovos"]

def preProcess(img):
    imgPre = cv.Canny(img_otsu2,90,140)
    kernel = np.ones((4,4),np.uint8)
    imgPre = cv.dilate(imgPre,kernel,iterations=2)
    imgPre = cv.erode(imgPre,kernel,iterations=1)
    hist = plt.hist(imgPre)
    return imgPre

def DetectarOvos(img):
    imgOvos = cv.resize(img,(224,224))
    imgOvos = np.asarray(imgOvos)
    imgOvosNormalize = (imgOvos.astype(np.float32)/127.0)-1
    prediction = model.predict(data)
    index = np.argmax(prediction)
    percent = prediction[0][index]

```

```

    classe = classes[index]
    return classe, percent

img = cv.resize(img,(640,480))
imgPre = preProcess(img)
contours,hi =
cv.findContours(imgPre,cv.RETR_EXTERNAL,cv.CHAIN_APPROX_NONE)
qtd = 0

for cnt in contours:
    area = cv.contourArea(cnt)

if area > 2000:
    x,y,w,h = cv.boundingRect(cnt)
    cv.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
    recorte = imgPre[y:y+h,x:x+w]
    classe,conf = DetectarOvos(recorte)
    cv.putText = (img,str(classe),cv.FONT_HERSHEY_SIMPLEX, 0.5,
(255,255,255),2)

for img_otsu2 in contours:
    qtd=qtd+1
if classe == "Ovos de mosquito Anopheles":
    print ("Foram detectados ",qtd, classe, "na imagem")
else:
    print ("Não foram detectados ovos do mosquito Anopheles na imagem , mas
foram detectados ", qtd, classe, "na imagem")

cv_imshow(img)
cv_imshow(imgPre)
cv.waitKey(1)

```

ANEXO A – IMAGENS ANALIZADAS PELO ALGORITMO

Imagem 1.png Fonte: Rocha, E. M. (2015). ESTUDO COMPARATIVO DA MICROBIOTA BACTERIANA CULTIVÁVEL ASSOCIADA À *Anopheles darlingi* Root, 1926, E SEU HÁBITAT. UEA, 13(3), 1576–1580.



Imagem 2.jpg Fonte: FERREIRA, M. U. Fundamentos biológicos da parasitologia humana. São Paulo: Manole, 2003. Acesso em: 26 mai. 2023.

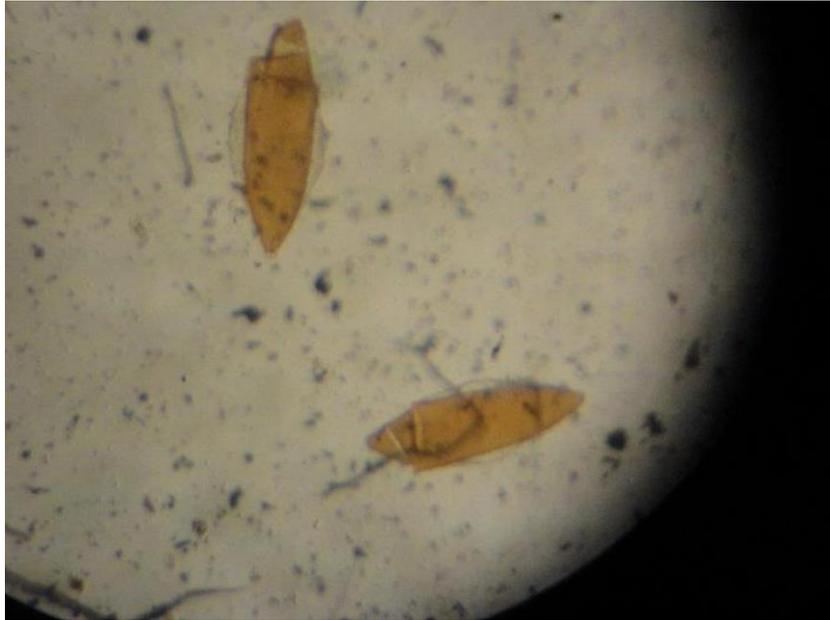


Imagem 3.jpg Fonte: CAMPBELL, D. Anopheles. Life Stages: Eggs. Disponível em: < <https://eol.org/pages/54865/articles>> Acesso em: 27 mai. 2023.



Imagem 4.jpg Fonte: WEEBLY. Anopheles Mosquito: Reproduction within the different phylums. Disponível em: < <https://biology02.weebly.com/anopheles-mosquito.html>> Acesso em: 26 mai. 2023.

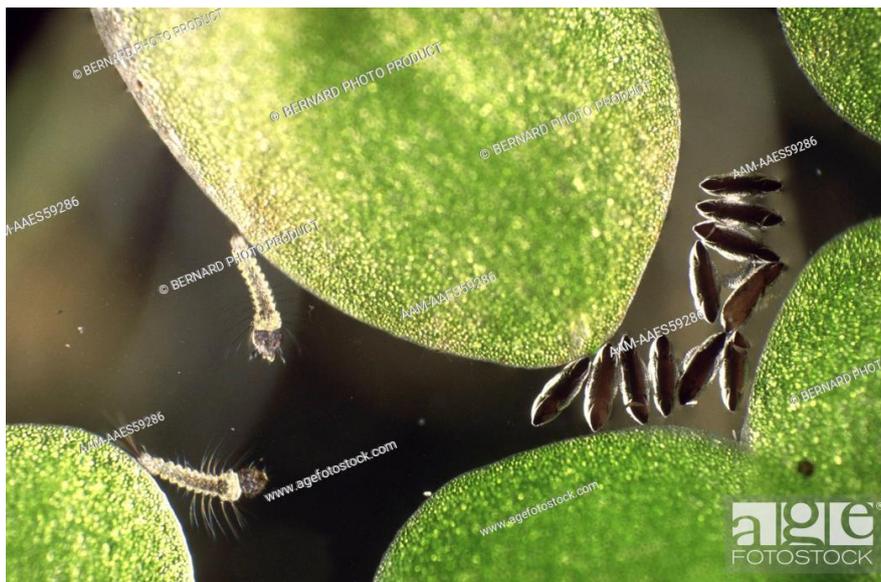


Imagem 5.jpg Fonte: BERNARD, G. Science Photo Library. Disponível em: <<https://www.sciencephoto.com/media/370217/view/anopheles-stephansi-mosquito-eggs-and-larva>> Acesso em: 16 ago. 2023.



Imagem 6.jpeg Fonte: Acervo Laboratório de Malário e Dengue/Inpa. Disponível em: <<https://www.gov.br/inpa/pt-br/assuntos/noticias/inpa-expoe-avancos-cientificos-e-tecnologicos-em-evento-alusivo-ao-dia-da-malaria-nas-americas>> Acesso em: 26 dez. 2022.

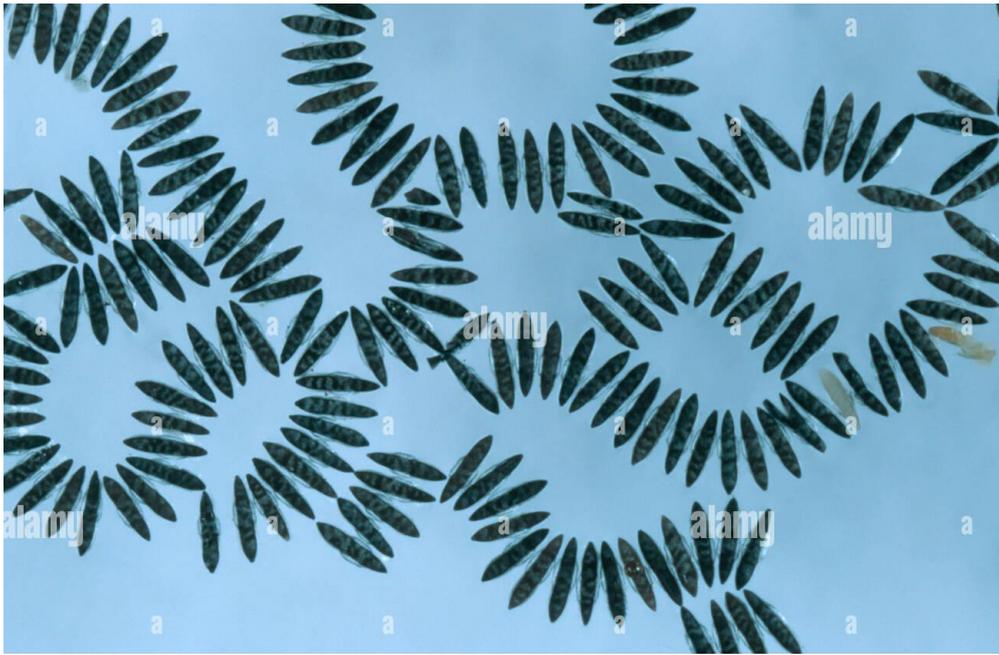


Imagem 7.jpg Fonte: ERITJA, R. Alamy Stock Photo. *Anopheles atroparvus* mosquito eggs floating in water viewed from above Disponível em: <<https://www.alamy.com/anopheles-atroparvus-mosquito-eggs-floating-in-water-viewed-from-above-image1067634.html>> Acesso em: 26 mai. 2023.



Imagem 8.jpg Fonte: SALADIN, K. What do mosquito eggs look like? Disponível em: <<https://www.quora.com/What-do-mosquito-eggs-look-like>> Acesso em: 26 dez. 2023.

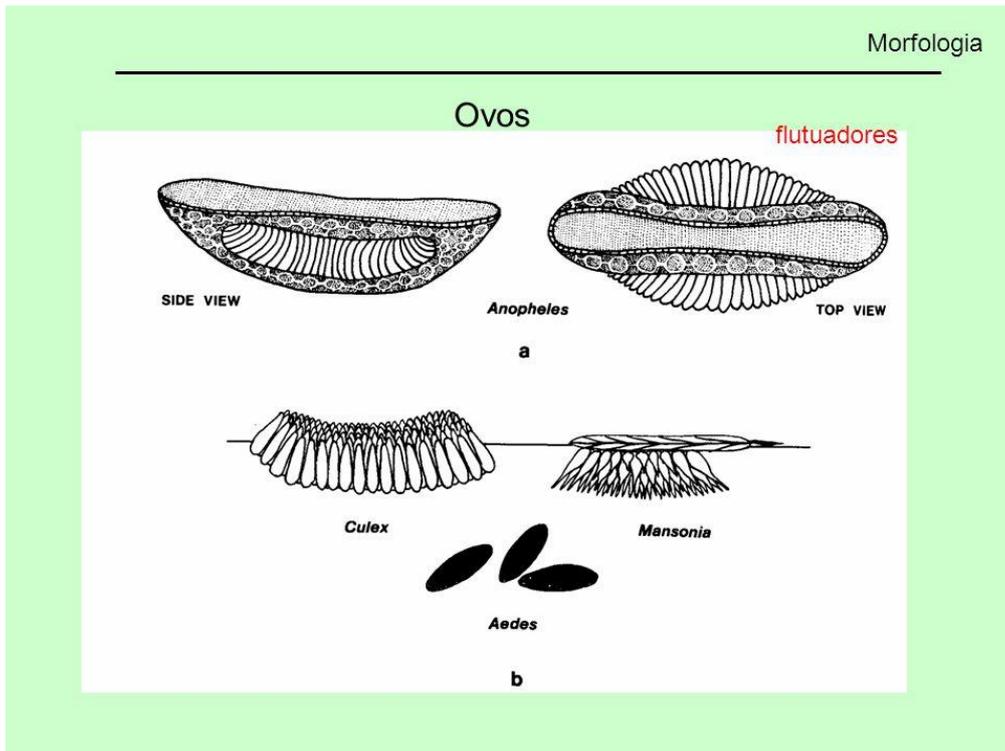


Imagem 9.jpg. Fonte: SABROSA, T. de A. Ordem Diptera. Disponível em: <<https://slideplayer.com.br/slide/5873563/>> Acesso em: 3 fev. 2023.



Imagem 10.png Fonte: Alomar, A. A., & Alto, B. W. (2022). Asian Malaria Mosquito *Anopheles stephensi* Liston, 1901 (Insecta : Diptera : Culicidae) 1. 1901(November), 1-7.



Imagem 11.jpg Fonte: Center for Disease Control and Prevention (CDC). Twenty-two eggs deposited by a female *Anopheles Stephensi* Mosquito. Disponível em: <<https://pixnio.com/pt/animais/inseto/mosquito/vinte-dois-ovos-depositados-femea-anopheles-stephensi-mosquito>> Acesso em: 26 dez. 2023.



Imagem 12.jpg Fonte: Library Book Collection. Alamy Stock Photo. Disponível em: <<https://www.alamy.com/animal-parasites-and-human-disease-insects-as-carriers-of-disease-medical-parasitology-430-mosquitoes-fig-194-eggs-of-anopheles-quadrimalatus-on-surface-of-water-x-13-after-howard-lay-their-eggs-singlj-fig-194-while-others-la-them-all-at-one-time-in-little-boat-shaped-rafts-called-egg-boats-the-individual-eggs-standing-upright-fig-195-the-fact-that-the-eggs-are-a-little-larger-at-the-lower-end-makes-the-whole-egg-boat-slightly-concave-thus-making-it-difficult-to-overturn-most-of-the-com-mon-mosciuitoes-of-temperate-climates-lay-their-eggs-on-the-open-surface-of-w-image236753628.html>> Acesso em: 26 mai. 2023.

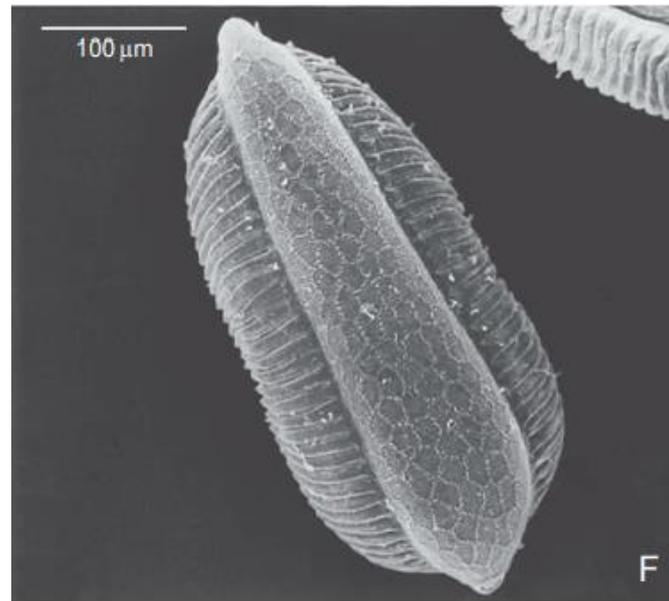


Imagem 13.png Fonte: Sallum, M. A. M., & Flores, D. C. (2004). Ultrastructure of the eggs of two species of *Anopheles* (*Anopheles*) Meigen (Diptera, Culicidae). *Revista Brasileira de Entomologia*, 48(2), 185–192. <https://doi.org/10.1590/s0085-56262004000200005>

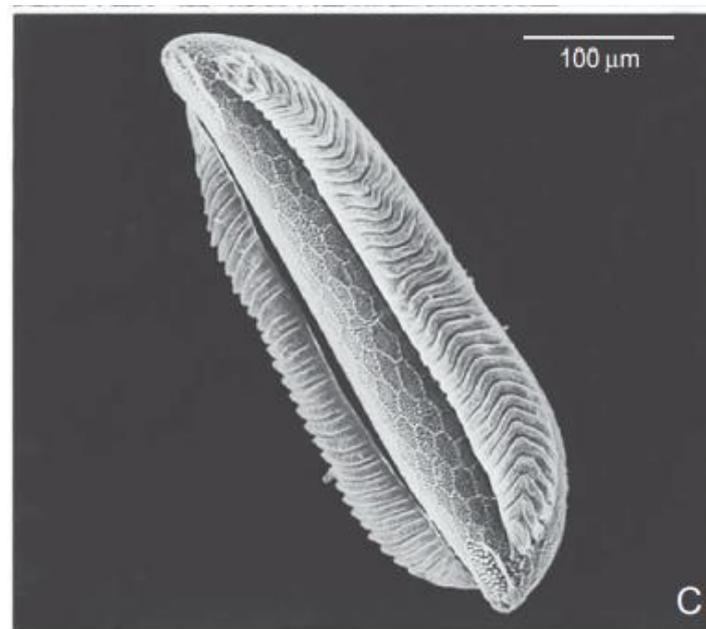


Imagem 14.png Fonte: Sallum, M. A. M., & Flores, D. C. (2004). Ultrastructure of the eggs of two species of *Anopheles* (*Anopheles*) Meigen (Diptera, Culicidae). *Revista Brasileira de Entomologia*, 48(2), 185–192. <https://doi.org/10.1590/s0085-56262004000200005>

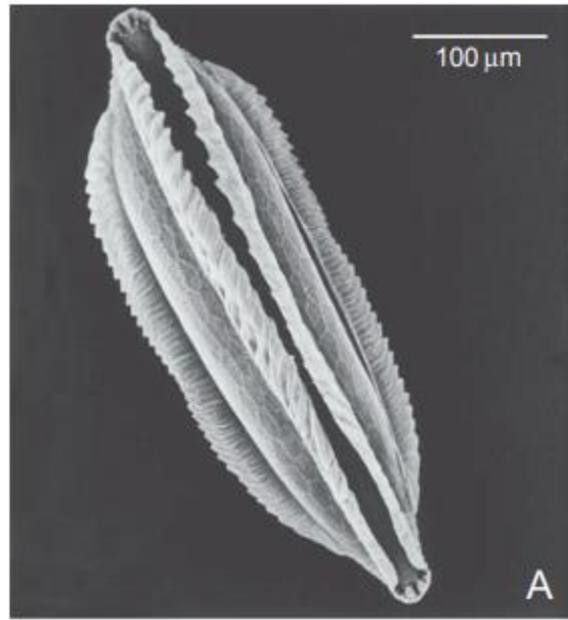


Imagem 15.png Fonte: Sallum, M. A. M., & Flores, D. C. (2004). Ultrastructure of the eggs of two species of *Anopheles* (*Anopheles*) Meigen (Diptera, Culicidae). *Revista Brasileira de Entomologia*, 48(2), 185–192. <https://doi.org/10.1590/s0085-56262004000200005>

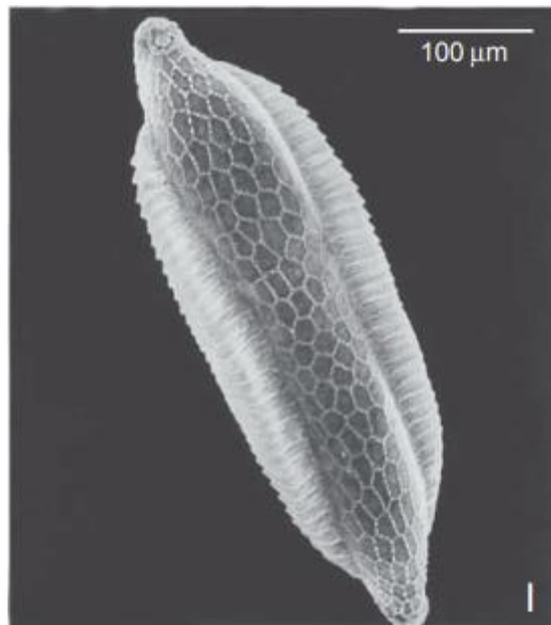


Imagem 16.png Fonte: Sallum, M. A. M., & Flores, D. C. (2004). Ultrastructure of the eggs of two species of *Anopheles* (*Anopheles*) Meigen (Diptera, Culicidae). *Revista Brasileira de Entomologia*, 48(2), 185–192. <https://doi.org/10.1590/s0085-56262004000200005>

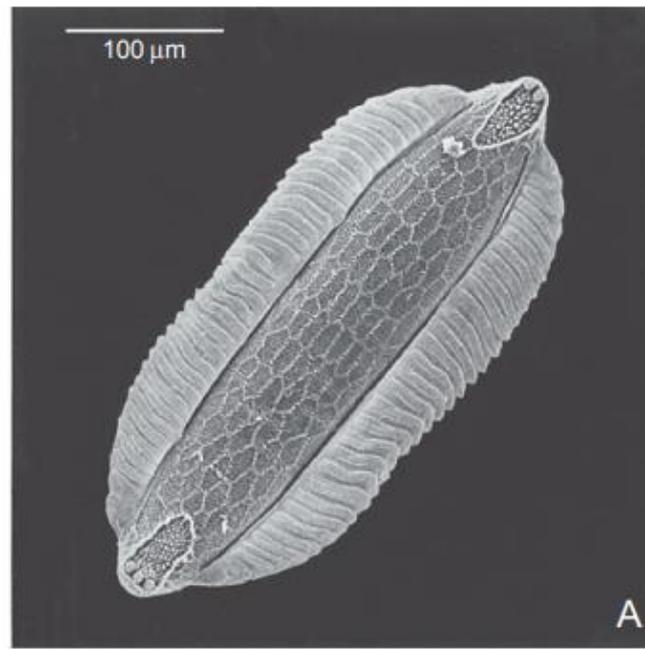


Imagem 17.png Fonte: Sallum, M. A. M., & Flores, D. C. (2004). Ultrastructure of the eggs of two species of *Anopheles* (*Anopheles*) Meigen (Diptera, Culicidae). *Revista Brasileira de Entomologia*, 48(2), 185–192. <https://doi.org/10.1590/s0085-56262004000200005>



Imagem 18.png Fonte: QUINTANA, T. Mosquito Biology and Control. Disponível em: < <https://www.slideserve.com/quintana-topaz/mosquito-biology-control> > Acesso em: 3 fev. 2023.

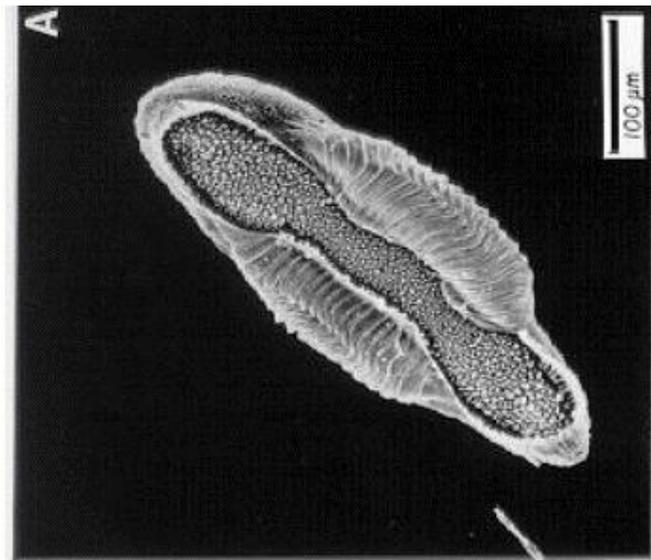


Imagem 19.png Fonte: USP. Família CULICIDAE Aspectos morfológicos e biológicos. (2018).

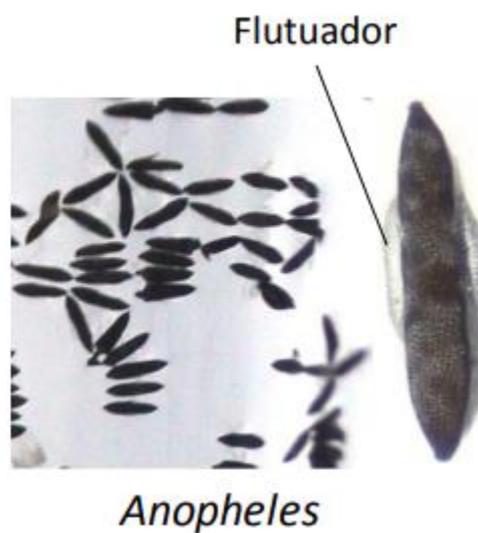
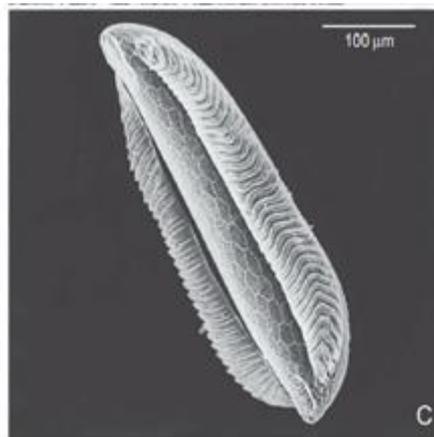


Imagem 20.png Fonte: CHAUVIN, G. Manuel de Formation à l'Entomologie du Paludisme. Disponível em: < <https://fr.readkong.com/page/manuel-de-formation-a-l-entomologie-du-paludisme-9319405> > Acesso em: 3 fev.2023.

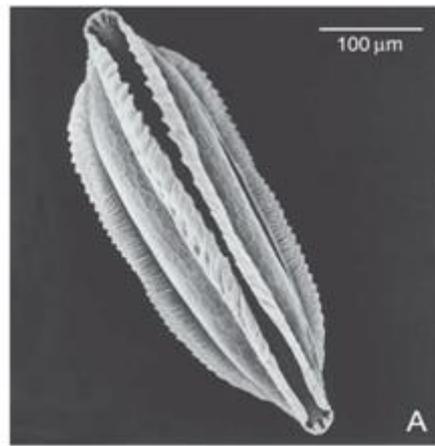
**ANEXO B – IMAGENS DE OVOS DO MOSQUITO ANOPHELES
UTILIZADAS NO MACHINE LEARNING**



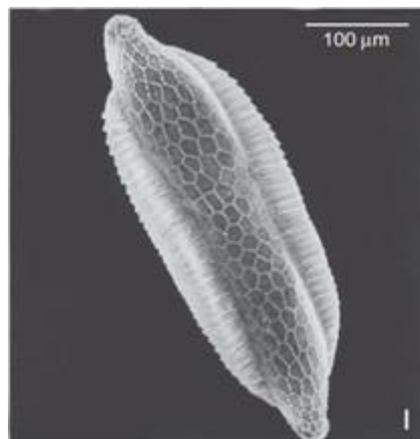
Fonte: Sallum, M. A. M., & Flores, D. C. (2004). Ultrastructure of the eggs of two species of *Anopheles* (*Anopheles*) Meigen (Diptera, Culicidae). *Revista Brasileira de Entomologia*, 48(2), 185–192. <https://doi.org/10.1590/s0085-56262004000200005>



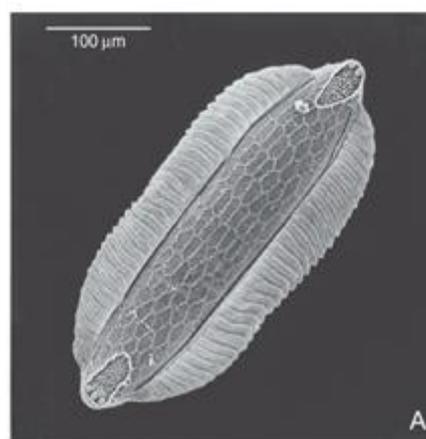
Fonte: Sallum, M. A. M., & Flores, D. C. (2004). Ultrastructure of the eggs of two species of *Anopheles* (*Anopheles*) Meigen (Diptera, Culicidae). *Revista Brasileira de Entomologia*, 48(2), 185–192. <https://doi.org/10.1590/s0085-56262004000200005>



Fonte: Sallum, M. A. M., & Flores, D. C. (2004). Ultrastructure of the eggs of two species of *Anopheles* (*Anopheles*) Meigen (Diptera, Culicidae). *Revista Brasileira de Entomologia*, 48(2), 185–192. <https://doi.org/10.1590/s0085-56262004000200005>



Fonte: Sallum, M. A. M., & Flores, D. C. (2004). Ultrastructure of the eggs of two species of *Anopheles* (*Anopheles*) Meigen (Diptera, Culicidae). *Revista Brasileira de Entomologia*, 48(2), 185–192. <https://doi.org/10.1590/s0085-56262004000200005>

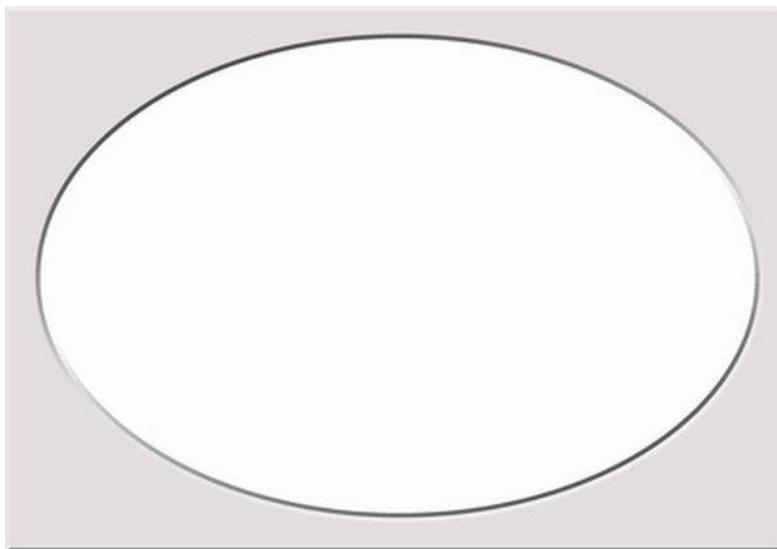


Fonte: Sallum, M. A. M., & Flores, D. C. (2004). Ultrastructure of the eggs of two species of *Anopheles* (*Anopheles*) Meigen (Diptera, Culicidae). *Revista Brasileira de Entomologia*, 48(2), 185–192. <https://doi.org/10.1590/s0085-56262004000200005>

**ANEXO C – IMAGENS DE NÃO-OVOS UTILIZADAS NO MACHINE
LEARNING**



Fonte: DELAUNAY, R. Windows Open Simultaneously 1st Part, 3rd Motif. The Solomon R. Guggenheim Foundation Peggy Guggenheim Collection, Venice, 1976. Disponível em: < <https://www.guggenheim.org/artwork/1024> > Acesso em: 16 ago. 2023.



Fonte: SWANSON, M. Como desenhar uma forma oval perfeita. Disponível em: < https://www.ehow.com.br/endurecer-papel-como_363456/ > Acesso em: 16 ago. 2023.



Fonte: Alphaville Artes e Leilões. Espelho oval com moldura dourada. Disponível em: <<https://www.galeriaalphaville.com.br/leiloes/142/lote/385>> Acesso em: 16 ago. 2023.



Fonte: Vexels. Ovo de Páscoa de linhas coloridas. Disponível em: <<https://br.vexels.com/png-svg/previsualizar/129858/ovo-de-pascoa-de-linhas-coloridas>> Acesso em: 16 ago. 2023.



Fonte: WEST, R. A generic white chicken egg. Disponível em:
<<https://www.flickr.com/photos/renwest/436827618>> Acesso em: 16 ago. 2023.