



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

RODRIGO EMERSON VALENTIM DA SILVA

**Contagem automática de ovos do mosquito *Aedes aegypti* utilizando métodos de
aprendizagem profunda e dispositivo de baixo custo**

Recife

2021

RODRIGO EMERSON VALENTIM DA SILVA

Contagem automática de ovos do mosquito *Aedes aegypti* utilizando métodos de aprendizagem profunda e dispositivo de baixo custo

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Inteligência Computacional.

Área de Concentração: Inteligência Computacional

Orientador: Leandro Maciel Almeida

Recife

2021

Catálogo na fonte
Bibliotecária Nataly Soares Leite Moro, CRB4-1722

S586c Silva, Rodrigo Emerson Valentim da
Contagem automática de ovos do mosquito *Aedes aegypti* utilizando métodos de aprendizagem profunda e dispositivo de baixo custo. – 2021.
88 f.: il., fig., tab.

Orientador: Leandro Maciel Almeida.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2021.
Inclui referências.

1. Inteligência computacional. 2. Redes neurais convolucionais. 3. Visão computacional. 4. *Aedes aegypti*. 5. Ovitampas. I. Almeida, Leandro Maciel (orientador). II. Título

006.31

CDD (23. ed.)

UFPE - CCEN 2022 – 41

Rodrigo Emerson Valentim da Silva

“Contagem automática de ovos do mosquito *Aedes aegypti* utilizando métodos de aprendizagem profunda e dispositivo de baixo custo”

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Inteligência Computacional

Aprovado em: 17/12/2021.

BANCA EXAMINADORA

Prof. Dr. Cleber Zanchettin
Centro de Informática / UFPE

Profa. Dra. Rosângela Maria Rodrigues Barbosa
Centro de Pesquisas Aggeu Magalhães / FIOCRUZ

Prof. Dr. Leandro Maciel Almeida
Centro de Informática / UFPE
(Orientador)

Decido este trabalho à minha família e minha namorada que foram porto seguro perante as dificuldades durante este percurso.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus por todas as realizações que tenho em minha vida.

Agradeço à minha família, por toda ajuda e aconselhamento que me deram durante a minha jornada.

Agradeço aos amigos da pós-graduação, por todo apoio durante esses anos, sem vocês eu não teria chegado até aqui: Cloves, Junior, Mailton, Daladier e Sthefanie.

Agradeço à minha namorada Patrícia Queiroz, por todo companheirismo, por sempre me apoiar e acreditar em mim.

Por fim, mas não menos importante, agradeço ao meu orientador, Leandro Maciel, por todas suas orientações durante o desenvolvimento deste trabalho.

RESUMO

O mosquito *Aedes aegypti* causa grandes transtornos ao ser humano, por ser o principal vetor de transmissão de arboviroses como: Dengue, Febre Amarela, Zika e Chikungunya que infectam mais de 400 milhões de pessoas a cada ano no mundo. Uma forma eficiente de evitar o crescimento no número de casos das arboviroses citadas é com programas de vigilância vetorial que ajuda no monitoramento de propagação de doenças causadas pelo *Ae. aegypti*. Entre os indicadores como quantidade de ovos, larvas, pupas ou quantidade fêmeas de mosquitos que a vigilância vetorial utiliza, os dispositivos utilizados para coleta de ovos de *Ae. aegypti* (ovitrapas), é um dos métodos mais eficientes na capacidade de detecção da presença de mosquitos mesmo em situação de baixa infestação. As ovitrapas consistem em recipientes escuros com entradas largas preenchidas parcialmente com água e uma palheta áspera de madeira instalada verticalmente em seu interior. Elas são colocadas em pontos estratégicos e depois de um tempo, a palheta é retirada do recipiente e levada para laboratório para ser possível fazer a contagem de ovos. Esta contagem é extremamente importante, pois as ovitrapas de regiões que apresentam o maior número de ovos recebem atenção especial de órgãos públicos em campanhas de combate ao mosquito. Entretanto, a contagem desses ovos em palhetas é feita manualmente, sendo uma atividade exaustiva e repetitiva para os técnicos que precisam realizar esta tarefa. Dado o contexto, neste trabalho foi construído um dispositivo que auxilia a contagem de ovos de *Ae. aegypti* com menos intervenção humana. Este dispositivo capta imagens da palheta, essas imagens são dadas como entrada para algoritmos de reconhecimento automatizado possibilitando o reconhecimento de ovos e conseqüentemente a sua contagem. Para alcançar os objetivos da automatização desta tarefa de contagem de ovos, foi utilizado o conceito de aprendizagem profunda com Redes Neurais Convolucionais (CNNs) que configuram o estado da arte em problemas de visão computacional. Para a utilização de CNNs, foi realizado um estudo de quais modelos poderiam ser adaptados ao problema de contagem de ovos, após escolhidos os modelos foi aplicado o processo de Transferência de Aprendizado destes modelos, visto que, os melhores modelos na literatura são treinados em grandes bases de dados e o que eles aprendem nessas bases pode ser reaproveitado em outras tarefas, de modo que se consegue treiná-los em outros conjuntos de dados, tornando o processo de treinamento mais rápido e mais efetivo em outros dados. Para treinar esses modelos foi construída uma base de dados própria com imagens de ovos de *Ae. aegypti*, visto que, uma das dificuldades encontradas em trabalhos da literatura que abordam a contagem automática

de ovos é falta de dados para treinar algoritmos para realizar esta tarefa. Os modelos de CNNs utilizados nesta dissertação apresentaram uma acurácia média no reconhecimento de ovos de 92%. Por fim, o estudo realizado servirá como base para construir uma ferramenta completa que auxilie todo o processo de contagem e tomada de decisão dos técnicos.

Palavras-chaves: redes neurais convolucionais; visão computacional; *Aedes aegypti*; ovitrampas

ABSTRACT

The *Aedes aegypti* mosquito causes great disturbances to humans, as it is the main vector of transmission of arboviruses such as: Dengue, Yellow Fever, Zika and Chikungunya that infect more than 400 million people each year in the world. An efficient way to prevent the increase in the number of cases of the arboviruses mentioned is with vector surveillance programs that helps the monitoring the spread of diseases caused by *Ae. aegypti*. Among the indicators such as number of eggs, larvae, pupae or number of female mosquitoes that vector surveillance uses are the devices used to collect *Ae. aegypti* eggs (ovitrap), it is one of the most efficient methods in the capacity of detecting the presence of mosquitoes even in situations of low infestation. Ovitrap consist of dark containers with wide inlets partially filled with water and a rough wooden reed installed vertically inside. They are placed at strategic points and after some time, the straw is removed from the pool and taken to the laboratory in order to count the eggs. This count is extremely important, as ovitrap from regions with high numbers of eggs receive special attention from public agencies in campaigns to combat the mosquito. However, counting eggs in straws is done manually, being an exhausting and repetitive activity for technicians who need to perform this task. Given the context, in this work, a device has been built to help to count *Ae. aegypti* eggs with less human intervention. This device captures images from the straw, these images are used as input to automated recognition algorithms enabling the recognition of eggs and consequently their counting. To achieve the goal of automating this task of egg counting, the concept of deep learning with Convolutional Neural Networks (CNNs) was used, which configures the state of the art in computer vision problems. In order to use CNNs, a study of which models could be adapted to the egg counting problem was carried out. After choosing the models, the Learning Transfer process of these models was applied, since the best models in the literature are trained in large databases and what they learn in these databases can be reused in other tasks, so that they can be trained in other data sets, making the training process faster and more effective on other data. In order to train these models, a database was built with images of *Ae. aegypti* eggs, since one of the difficulties found in studies in the literature that address the automatic counting of eggs is the lack of data to train algorithms to perform this task. The CNN models used in this work achieved an average accuracy in egg recognition of 92%. Finally, the study carried out will serve as the basis for building a complete tool that will assist the entire counting and decision-making process of technicians

Keywords: convolutional neural networks; computer vision; *Aedes aegypti*; ovitraps.

LISTA DE FIGURAS

Figura 1 – Número de casos e óbitos por Dengue, Chikungunya, Zika no Brasil de 2008 a 2019.	18
Figura 2 – Fases do ciclo de vida de <i>Ae. aegypti</i>	23
Figura 3 – Ovitrapa. Ovitrapa é uma armadilha composta por uma palheta de madeira e um recipiente escuro com água.	27
Figura 4 – Técnico fazendo a contagem de ovos com auxílio de um estereomicroscópio. Esta contagem é feita de forma manual.	27
Figura 5 – Imagem de microscopia de uma palheta de ovitrapa com presença de ovos.	27
Figura 6 – Passos fundamentais de processamento em um sistema de visão computacional.	29
Figura 7 – Modelo não-linear de um neurônio	34
Figura 8 – Modelo Perceptron Multicamadas.	36
Figura 9 – Exemplo de uma rede neural convolucional. Na Figura é possível observar uma imagem do dígito zero entrando numa CNN que a faz o processo convolução, <i>pooling</i> e classificação de qual dígito é a imagem, neste caso, é esperado que a CNN classifique como dígito zero.	38
Figura 10 – Diferentes métodos de agregação baseados em FPN. A principais diferenças são desempenho e a eficiência. A PANet atinge melhor precisão do que a FPN e a NAS-FPN, entretanto tem mais parâmetros. Já BiFPN otimiza esses modelos com conexões de escala cruzada.	40
Figura 11 – Arquitetura geral da EfficientDet. Utilizam redes pré-treinadas no ImageNet como rede de <i>backbone</i> . BiFPN Funciona como extrator de recursos, por fim, os recursos são combinados e utilizados pela rede para fazer a predição de classes.	41
Figura 12 – Arquitetura da YOLOv4. O <i>input</i> é a imagem passada para a rede, em seguida, essa imagem entra no <i>backbone</i> que funciona como extrator de características, a parte de <i>neck</i> é empregada para fazer a fusão das melhores características. Por fim, nos dois últimos passos são efetuadas as predições.	43
Figura 13 – Resultados da YOLOv4 na competição MS COCO comparada as redes YOLOv3, EfficientDet, ATSS e CenterMask.	43

Figura 14 – Arquitetura do modelo de rede neural RFCN para detecção de objetos. Da esquerda para a direita: O modelo recebe uma imagem como entrada e aplica seu processamento para identificar o objeto.	45
Figura 15 – fluxo	56
Figura 16 – Dispositivo de digitalização de palhetas de ovitrampas desenvolvido neste trabalho sendo mostrado de três ângulos.	57
Figura 17 – Imagem coletada com o dispositivo digitalizador de palhetas. Na imagem é possível ver nitidamente os ovos.	58
Figura 18 – Interface do <i>Software</i> . Nesta interface o usuário pode acionar a máquina para digitalizar a palheta e fazer a contagem de ovos automática com o modelo de CNN selecionado.	59
Figura 19 – Fluxo para utilização do dispositivo para digitalização, contagem e revisão de resultados.	60
Figura 20 – Palheta e suas dimensões.	61
Figura 21 – Exemplo de anotação de imagem. Ao centro é possível observar o processo de criação de caixas nos ovos. Cada caixa marca as coordenadas x e y, tais informações são importantes para os modelos de detecção buscarem especificamente nas imagens o que eles tem que aprender e onde.	63
Figura 22 – Imagem original antes da normalização de histograma. Como é possível observar, a imagem é bastante escura dificultando a visualização do ovo na parte inferior.	64
Figura 23 – Imagem após a normalização de histograma. Já nesta imagem é possível observar que o ovo aparece na parte inferior. Tal melhoria, já faz os modelos reconhecerem o padrão do ovo.	64
Figura 24 – Processo de aumento de imagens a partir de uma imagem original, são feitas novas imagens aplicando filtros e rotações. Nota-se que o ovo presente na imagem vai se deslocando a medida que as aplicações de aumento de dados são aplicadas.	65
Figura 25 – Histograma do número de ovos em imagens. As imagens são da base de dados construída neste trabalho. Grande parte das imagens apresenta de 1 a 10 ovos.	67

Figura 26 – Exemplo de detecção de uma CNN. No exemplo a rede aplica a predição na imagem e cria duas caixas, uma, no homem e outra no cachorro. Além das caixas a rede também mostra a probabilidades para cada caixa.	68
Figura 27 – Início de treinamento do modelo YOLOv4 que começa criando a quantidade de caixas possível, pois ainda não aprendeu a reconhecer os ovos.	69
Figura 28 – Fim do treinamento do modelo. Na Figura é possível notar que agora o modelo marca apenas os ovos que aparecem na imagem, ou seja, o modelo aprendeu a reconhecer corretamente o padrão do ovo.	70
Figura 29 – Gráfico com as contagens entre humanos e modelos de CNN.	76
Figura 30 – Erro de detecção. Os números 1, 2 e 3 são ovos, entretanto o ovo 3 não foi detectado pela CNN.	77
Figura 31 – Correção de detecção. Os números 1, 2 e 3 são ovos. O ovo 3 agora foi detectado corretamente pela CNN.	77
Figura 32 – Exemplo de falso positivo na detecção de um ovo representado pelo retângulo vermelho.	78
Figura 33 – Tempo médio em segundos necessários para cada modelo contar uma palheta. Em azul o tempo do YOLOv4, em laranja o do EfficientDet e em verde o do RFCN.	80

LISTA DE TABELAS

Tabela 1 – Trabalhos Relacionados	50
Tabela 2 – Divisão da base de dados entre dados de treinamento, validação e teste. Cada linha da tabela apresenta a quantidade de imagens no conjunto de dados e a quantidade total de ovos que tem nas imagens.	66
Tabela 3 – Matriz de Confusão.	71
Tabela 4 – Resultado dos modelos nos dados de treino, validação e teste.	74
Tabela 5 – Comparação entre a contagem efetuada por pessoas e a contagem feita por modelos de CNNs.	76
Tabela 6 – Resultado do teste de normalidade com Shapiro-Wilk. Cada linha apresenta o contador, estatística de teste e o resultado do p-valor.	79
Tabela 7 – Comparação de p-valores entre cada par de contadores. Os resultados em vermelho são os que são menores que 0.05.	79

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizagem de Máquina
BiFPN	Weighted Bi-directional Feature Pyramid Network
BP	<i>Back-Propagation</i>
CNN	Convolutional Neural Network
EfficientDet	Efficient Object Detection
FN	Falso Negativo
FP	Falso Positivo
FPN	Feature Pyramid Network
GPU	Graphics Processing Unit
IA	Inteligência Artificial
ILSVRC	ImageNet Large Scale Visual Recognition
MLP	<i>Multilayer Perceptron</i>
PDI	Processamento Digital de Imagem
RFCN	Region-based Fully Convolutional Network
RNA	Rede Neural Artificial
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

LISTA DE SÍMBOLOS

γ Letra grega Gama

\in Pertence

δ Delta

θ Teta

σ Sigma

μ Mi

SUMÁRIO

1	INTRODUÇÃO	18
1.1	OBJETIVOS	20
1.2	PRINCIPAIS CONTRIBUIÇÕES	20
1.3	ORGANIZAÇÃO DO DOCUMENTO	21
2	REFERENCIAL TEÓRICO	22
2.1	MOSQUITO <i>AE. AEGYPTI</i>	22
2.1.1	Transmissão de doenças	24
2.1.2	Vigilância do vetor	24
2.1.3	Ovitrapa e contagem manual	25
2.2	SISTEMA DE VISÃO COMPUTACIONAL	28
2.3	INTELIGÊNCIA ARTIFICIAL E APRENDIZADO DE MÁQUINA	30
2.4	REDES NEURAIS ARTIFICIAIS	32
2.4.1	Redes Neurais Profundas Convolucionais	37
2.4.2	EfficientDet	39
2.4.3	YOLOv4	42
2.4.4	Region-based Fully Convolutional Network (RFCN)	44
2.5	CONCLUSÃO DO CAPÍTULO	46
3	TRABALHOS RELACIONADOS	47
3.1	CONCLUSÃO DO CAPÍTULO	55
4	MATERIAIS E MÉTODOS	56
4.1	FLUXO DE DESENVOLVIMENTO DA PESQUISA	56
4.2	DISPOSITIVO DE <i>HARDWARE</i> DESENVOLVIDO PARA CAPTURA DE DADOS	57
4.3	BASE DE DADOS DE IMAGENS	61
4.3.1	Aquisição das palhetas	61
4.3.2	Anotação das imagens	62
4.3.3	Pré-processamento das imagens	63
4.3.4	Divisão da base de dados	66
4.4	MODELOS DE CNNs UTILIZADOS NESTE TRABALHO	67
4.5	TREINAMENTO DOS MODELOS CNNs UTILIZADAS	69

4.6	MÉTRICAS UTILIZADAS PARA AVALIAR OS RESULTADOS DOS MODELOS DE CNN ESCOLHIDOS	70
4.6.1	Testes estatísticos utilizados	72
4.7	CONCLUSÃO DO CAPÍTULO	73
5	RESULTADOS DOS EXPERIMENTOS	74
5.1	DESEMPENHO NA BASE DE DADOS	74
5.2	DESEMPENHO DOS MODELOS COMPARADOS AOS TÉCNICOS	75
6	CONCLUSÃO E TRABALHOS FUTUROS	81
	REFERÊNCIAS	84

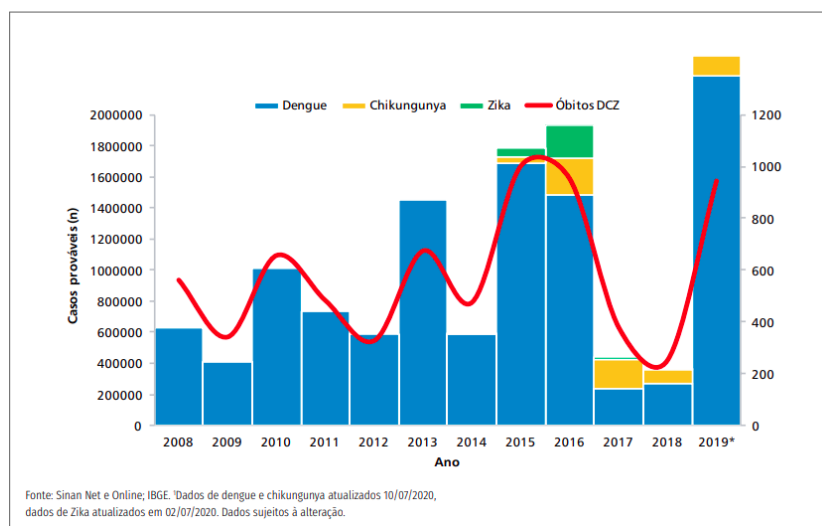
1 INTRODUÇÃO

Popularmente conhecido como mosquito da dengue, *Ae. aegypti* (nomenclatura taxonômica) causa grandes transtornos ao sistema público de saúde, por ser o principal vetor de transmissão de arboviroses como: Dengue, Febre Amarela, Zika e Chikungunya que infectam mais de 400 milhões de pessoas a cada ano (LETA et al., 2018).

No Brasil, as arboviroses mencionadas criam um cenário complexo com a circulação simultânea dos quatro sorotipos dos vírus da dengue (DENV-1, DENV-2, DENV-3 e DENV-4). A partir de 2014, este cenário se agrava com a circulação do vírus Chikungunya (CHIKV) e, em 2015, com o vírus Zika (ZIKV). Mesmo com todos os esforços realizados pela União, Estados e Municípios no Sistema Único de Saúde (SUS), a ocorrência de grandes epidemias de dengue e da chikungunya tem sido cada vez mais recorrente, principalmente em grandes centros urbanos (BRASIL, 2020).

No período de 2008 a 2019, foram notificados no Brasil cerca de 11,6 milhões de casos de dengue, chikungunya e zika (DCZ). Neste mesmo período, foram confirmados 7.043 óbitos por essas doenças mencionadas (BRASIL, 2020). A Figura 1 apresenta o número de casos de dengue, chikungunya e zika. Como podemos observar na figura, a dengue mostra uma maior relevância entre as demais doenças em decorrência do seu número alto de casos.

Figura 1 – Número de casos e óbitos por Dengue, Chikungunya, Zika no Brasil de 2008 a 2019.



Fonte: (BRASIL, 2020)

Portanto, a circulação dos arbovírus DENV, ZIKV e CHIKV causa um importante impacto na morbimortalidade da população suscetível, pois com o aumento do número de indivíduos

infectados, há também a necessidade do aumento da demanda pelos serviços de saúde (BRASIL, 2020).

Dada essa problemática, uma forma eficiente de evitar o crescimento no número de casos de arboviroses é com implemento de programas de vigilância vetorial e epidemiológica que ajudam na definição de fatores e populações de risco através de indicadores, além de auxiliar no monitoramento de propagação de doenças, detecção/predição de epidemias, planejamento das atividades e a avaliação dos programas de controle (ACIOLI et al., 2006).

A vigilância epidemiológica para *Ae. aegypti* utiliza como indicador a quantidade de ovos, larvas, pupas ou quantidade de mosquitos adultos visando estimar a população vetora em um determinado local (ACIOLI et al., 2006). Entre esses indicadores, a contagem de ovos em palhetas de ovitrampas tem sido um dos métodos mais eficientes, superior até a pesquisa larvária na capacidade de detecção da presença de mosquitos mesmo em situação de baixa infestação, além de ter baixo custo de confecção (ACIOLI et al., 2006).

A desvantagem desse indicador é o fato do processo ser feito por técnicos de forma manual, tornando assim o procedimento suscetível a diferentes categorias de erros humanos na contagem, mesmo sendo feito por técnicos experientes. Geralmente a contagem de ovos é feita com o auxílio de uma lente de aumento ou microscópios ópticos e contadores de mão, tornando uma atividade bastante exaustiva até para profissionais mais treinados.

Uma forma de ajudar os técnicos a conseguirem contar essas palhetas de maneira mais precisa é com a criação de um dispositivo que automatize o processo de digitalização e conte automaticamente os ovos nas palhetas. Entretanto, esta também não é uma tarefa fácil, pois há grandes desafios que precisam ser considerados para construir um dispositivo eficiente, entre eles, a pouca disponibilidade de dados públicos relacionados a contagem automática de ovos do mosquito *Ae. aegypti*, falta de equipamentos digitais que consigam captar imagens dos seus ovos de forma fácil, de baixo custo e aplicável em laboratórios. Esses desafios são evidenciados nos trabalhos que abordam a contagem automática dos ovos de *Ae. aegypti*.

Dado o contexto, o desafio deste projeto é criar um dispositivo que auxilie a contagem de ovos de *Ae. aegypti* com o mínimo de intervenção humana possível, e com uma boa taxa de acerto. Esse dispositivo pode possibilitar que a vigilância epidemiológica de *Ae. aegypti* seja mais célere no monitoramento de transmissão de doenças, detectando epidemias e planejando atividades de programas de controle do mosquito. Diminuindo, assim, os prejuízos causados pelas doenças transmitidas pelo *Ae. aegypti*.

Para a construção de um dispositivo capaz de realizar a contagem automática de ovos de

Ae. aegypti, esse trabalho recorre à aprendizagem profunda com Redes Neurais Artificiais que é um tópico de Inteligência Artificial que vem trazendo avanços para a sociedade na resolução de diferentes problemas que envolvem habilidades visuais. Alguns exemplos de uso de redes neurais profundas são: controle de carros autônomos (POSSATTI et al., 2019), reconhecimento de faces (GUO; ZHANG, 2019), contagem plantas (KITANO et al., 2019) e contagem de frutas (CHEN et al., 2017). Dada a variedade de aplicações, este trabalho experimenta a adaptação dessa tecnologia para contagem automática de ovos de *Ae. aegypti*.

1.1 OBJETIVOS

O objetivo geral da presente pesquisa é desenvolver um dispositivo composto de um *hardware* e um *software* para contagem automática de ovos do mosquito *Ae. aegypti* em ovitrampas, visto que, o principal meio de contagem de seus ovos é manual com manuseio de ferramentas simples como lupas e contadores de mão.

Para o alcance do objetivo geral, são definidos os seguintes objetivos específicos:

- a) Construir um dispositivo de *hardware* para digitalização de palhetas de ovitrampas;
- b) Criar uma base de dados com imagens de palhetas de campo;
- c) Treinar diferentes modelos de Redes Neurais artificiais para o reconhecimento de ovos em imagens;
- d) Analisar se há discrepância na contagem de palhetas entre humanos;
- e) Criar um *software* para controle do dispositivo e contagem de ovos com Redes Neurais;

1.2 PRINCIPAIS CONTRIBUIÇÕES

- a) Criação de um dispositivo de *hardware* específico para a coleta de dados ou captura de imagens;
- b) Construção de uma base de dados robusta para fomentar o treinamento de métodos de aprendizagem profunda;
- c) Utilização de métodos com Redes Neurais artificiais devido ao alto desempenho em tarefas de reconhecimento de padrões em imagens e;

- d) Criação de um protótipo de *software* para apoiar técnicos na função de contagem de ovitrampas;

1.3 ORGANIZAÇÃO DO DOCUMENTO

O trabalho está estruturado da seguinte forma: o Capítulo 2 são expostos os conceitos que são necessários para a fundamentação do trabalho. O capítulo 3 traz uma revisão da literatura de trabalhos e contagem automática. O Capítulo 4 apresenta os materiais e métodos. O capítulo 5 mostra os resultados. Por fim, no Capítulo 6 são realizadas as considerações finais.

2 REFERENCIAL TEÓRICO

Este capítulo visa contextualizar o mosquito *Ae. aegypti* e os problemas por ele causados na sociedade. Objetiva, ainda, apresentar conceitos importantes para o entendimento do trabalho e mostrar o estado da arte no desenvolvimento de ferramentas para contagem automática de ovos do mosquito *Ae. aegypti*.

2.1 MOSQUITO *AE. AEGYPTI*

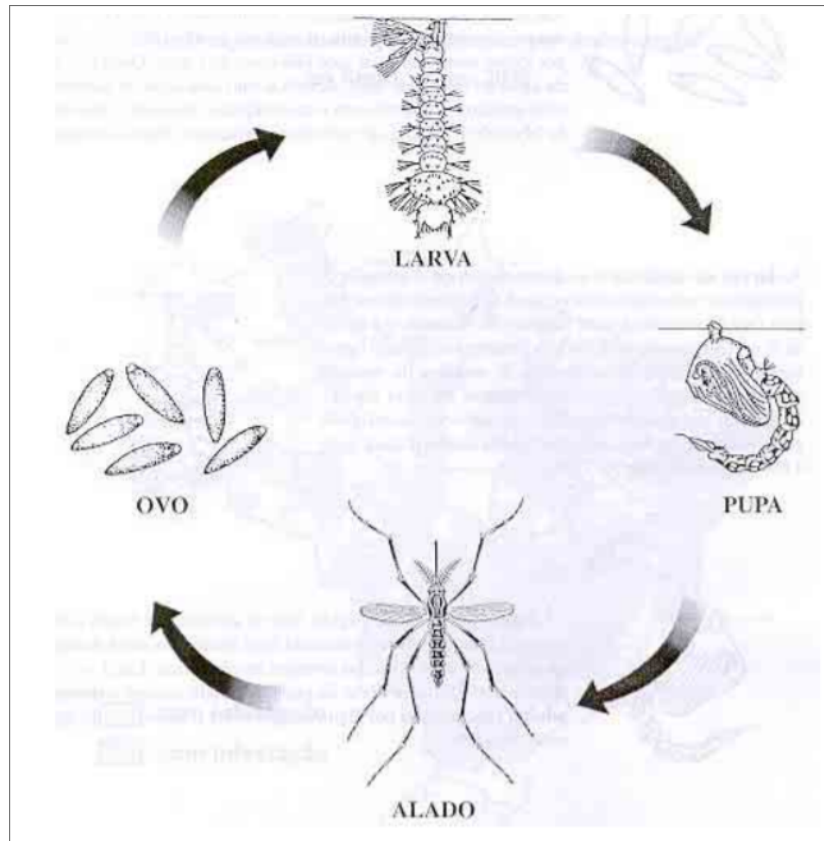
Ae. aegypti é um mosquito diurno, de cor preta, que possui listras e manchas brancas sendo adaptado ao ambiente urbano (TAVEIRA et al., 2001). O mosquito se desenvolve em água parada e não necessariamente limpa, acumulada em recipientes artificiais ou naturais (ARDUINO; ÁVILA, 2015).

As fêmeas do mosquito *Ae. aegypti* causam grandes transtornos ao sistema público de saúde, por serem os principais vetores de transmissão dos vírus Dengue, Febre Amarela, Zika e Chikungunya que infectam mais de 400 milhões de pessoas a cada ano (LETA et al., 2018).

Sua origem pode ser rastreada na África, onde se adaptou ao ambiente urbano e hoje ocorrendo em áreas tropicais e subtropicais do planeta (SOGHIGIAN et al., 2020). Seu comportamento sinantrópico e antropofílico está relacionado à sua adaptação a criadouros artificiais nos quais as fêmeas põem seus ovos (SOUZA, 2016). Por ser um mosquito bastante próximo do ser humano, esses ovos são postos aos redores de residências em recipientes com água tais como: vasilhames de plástico, garrafas, pneus, jarros de plantas, entre outros (SOUZA, 2016). Tal facilidade de encontrar recipientes como estes, facilita a contaminação da população por doenças transmitidas pelo mosquito.

De acordo com Taveira et al. (2001) o ciclo evolutivo do mosquito, como demonstra a Figura 2, consiste em duas fases:

- Aquática — Nesta fase, ocorrida na água, acontecem os estágios imaturos: ovo, larva e pupa.
- Alada – Essa fase ocorre em ambientes terrestre e aéreo e é constituída pelo mosquito adulto.

Figura 2 – Fases do ciclo de vida de *Ae. aegypti*

Fonte: Taveira et al. (2001)

A evolução do ovo para o mosquito adulto, como observa Taveira et al. (2001), ocorre conforme a temperatura e disponibilidade de alimento, de modo que normalmente, em condições ideais (ambiente com temperatura elevada e alimentação adequada) a evolução ocorra completamente entre 9 e 13 dias.

Os ovos consistem na primeira etapa do ciclo de vida de *Ae. aegypti* e são o objeto de estudo da presente pesquisa. Eles possuem coloração preta, são extremamente pequenos e só são reconhecidos a partir de um exame cuidadoso. Ademais, são extremamente resistentes, capazes de se manterem viáveis por vários meses até mesmo em local seco (TAVEIRA et al., 2001).

Por essa capacidade de resistência, o número de casos é elevado, como mostra o Boletim Epidemiológico fornecido pelo Ministério da Saúde Brasileiro, até a Semana Epidemiológica 18 de 2021 foram notificados 279.743 casos prováveis (taxa de incidência de 132,1 casos por 100 mil hab.).

A seguir será discutido sobre as doenças transmitidas pelo mosquito, a vigilância do vetor, a pesquisa larvária, coleta de mosquitos e armadilha de oviposição ou Ovitampa

2.1.1 Transmissão de doenças

O mosquito *Ae. aegypti* é considerado o principal vetor de transmissão dos quatro sorotipos do vírus dengue em circulação no Brasil (DENV-1, DENV-2, DENV-3 e DENV-4), além de transmitir os arbovírus Chikungunya e Zika, bem como a Febre Amarela (DONALISIO; FREITAS; ZUBEN, 2017).

Essas doenças citadas apresentaram um aumento global relevante nos últimos anos, sendo os principais fatores a dispersão mais rápida e geograficamente mais extensiva dos vírus em razão do crescimento intensivo dos sistemas de transporte globais, adaptação dos vetores à urbanização crescente, a incapacidade de conter a população de mosquitos e as alterações em fatores ambientais (TEICH; ARINELLI; FAHHAM, 2017).

Como observam Gregianini et al. (2017) o Brasil apresenta ótimas condições ambientais que permitem a permanência e disseminação de mosquitos vetores, como *Ae. aegypti*. Por isso, é necessário o investimento em serviços públicos que permitam o combate a prevenção das arboviroses. Pensando nisso, foram criados programas como: o Programa Nacional de Controle da Dengue (PNCD), que tem como foco principal o controle de *Ae. aegypti* pela população e por profissionais com o apoio do governo e o Programa Nacional de Apoio ao Combate às Doenças Transmitidas pelo Aedes (Pronaedes), que financia projetos de combate à proliferação das doenças transmitidas pelo vetor (FIGUEIRÓ et al., 2010).

As doenças causadas por arboviroses trazem impactos sociais e econômicos, visto que os pacientes em fase aguda das doenças podem apresentar sintomas que duram por semanas ou meses, comprometendo atividades laborais. No aspecto econômico, para Teich, Arinelli e Fahham (2017), o Brasil gastou cerca de R\$1,5 bilhão em 2016 para combater o vetor *Ae. aegypti*. Sendo que os maiores repasses federais foram destinados aos estados de São Paulo, Minas Gerais, Bahia e Rio de Janeiro, por se tratarem das UFs com maior população no País. As UFs com menor população, como Pará, Ceará, Pernambuco e Maranhão, integrantes das regiões Norte-Nordeste, vêm, em seguida, como as UFs a receber maior repasse para combater o vetor no país.

2.1.2 Vigilância do vetor

Atualmente a principal forma de evitar o crescimento no número de casos de arboviroses é com programas de vigilância vetorial e epidemiológica que ajuda na definição de fatores e

populações de risco através de indicadores, além de auxiliar no monitoramento de transmissão de doenças, detecção/predição de epidemias, planejamento das atividades e a avaliação dos programas de controle (ACIOLI et al., 2006).

A vigilância epidemiológica para mosquitos utiliza como indicador a quantidade de larvas, ovos, pupas ou quantidade adultos com o objetivo de estimar a população vetora em determinados locais (ACIOLI et al., 2006).

Segundo Braga e Valle (2007) a pesquisa larvária, ou seja, a coleta de larvas, é importante para verificar o impacto que as estratégias básicas para controle da doença causam. No entanto, esta técnica não permite medir a quantidade de mosquitos adultos, sendo ineficaz para estimar o risco de transmissão da doença.

Braga e Valle (2007) também salienta que a coleta de mosquitos é uma metodologia utilizada para realizar a vigilância epidemiológica. A coleta de mosquitos adultos é realizada apenas em situações específicas nos programas de controle de dengue, uma vez que é demorada e possui altos custos de operacionalização.

Ademais, ainda de acordo com Braga e Valle (2007) as informações obtidas com a coleta de mosquitos adultos possuem um valor limitado para uma avaliação de risco de transmissão, visto que não é possível relacionar o número de insetos coletados com os números absolutos de adultos, tendo em vista que os mosquitos desenvolvidos podem repousar tanto dentro como fora de residências e em locais pouco acessíveis. Sendo assim, o número de mosquitos coletados utilizando esta metodologia apenas representa uma estimativa do total. Outro obstáculo ao usar esta metodologia para avaliar o risco de transmissão é que não tem como relacionar o número de mosquitos adultos e a transmissão, pois a correlação entre o número de vetores coletados e o número de humanos na área de coleta não é suficiente para averiguar o risco (BRAGA; VALLE, 2007).

Outro método para definir a densidade populacional de *Ae. aegypti* é com armadilha de oviposição (Ovitampa) que faz a estimativa populacional do mosquito por meio do número de ovos em palhetas de ovitrampas. O funcionamento das ovitrampas e a contagem de ovos são explicadas na próxima Seção.

2.1.3 Ovitampa e contagem manual

O *Ae. aegypti* é um mosquito muito bem-adaptado ao ambiente urbano, ele se utiliza de todos os reservatórios artificiais próximos a residências e aos arredores para pôr seus ovos,

apresentando assim grandes desafios para o controle de crescimento de sua população em determinadas regiões (WILKE et al., 2020). Os locais mais comuns onde as fêmeas de *Ae. aegypti* põem seus ovos são: tanques de armazenamento de água, vasilhames, barris, pneus usados, latas, garrafas e vasos de plantas. Baseado neste cenário, foi desenvolvido por Fay, Eliason et al. (1966) uma armadilha de oviposição, ou ovitrampa que é um dispositivo utilizado para coleta de ovos do mosquito *Ae. aegypti* e é um dos métodos mais eficientes, superior até a pesquisa larvária na capacidade de detecção da presença de mosquitos mesmo em situação de baixa infestação, além de ter baixo custo de confecção (ACIOLI et al., 2006).

As ovitrampas consistem em recipientes escuros com entradas largas preenchidas parcialmente com água e uma palheta áspera de madeira instalada verticalmente em seu interior (GOMES et al., 2003). Elas são colocadas em pontos estratégicos de residências e depois de um tempo, a palheta é retirada do recipiente e levada para laboratório para ser possível fazer a contagem de ovos.

Nos países tropicais onde a dengue é uma doença endêmica, a ovitrampa é o principal meio para avaliar o impacto de medidas de controle visando impedir a dispersão da população de *Ae. aegypti* em uma determinada região. As ovitrampas podem também ser usadas para determinar a densidade das populações de *Ae. aegypti* em regiões onde medidas de controle foram aplicadas (SANTOS, 2021).

A contagem de palhetas é feita de forma manual com auxílio de um contador de mão e ferramentas como lupa e estereomicroscópio, sendo assim, uma atividade que exige muita atenção e esforço repetitivo, pois são colhidas e contadas centenas de palhetas. As Figuras 3, 4 e 5 mostram o processo de contagem manual. Na Figura 3 é apresentado uma ovitrampa completa composta por um recipiente escuro e uma palheta de madeira no centro. A Figura 4 mostra um técnico contando uma palheta da ovitrampa com auxílio de um estereomicroscópio. Por fim, a Figura 5 mostra uma imagem de microscopia de uma palheta onde é possível ver os ovos de coloração escura. Algumas vezes os técnicos utilizam apenas uma lupa e uma luz para ver os ovos na palheta ao invés de usar um microscópio.

Figura 3 – Ovitampa. Ovitampa é uma armadilha composta por uma palheta de madeira e um recipiente escuro com água.



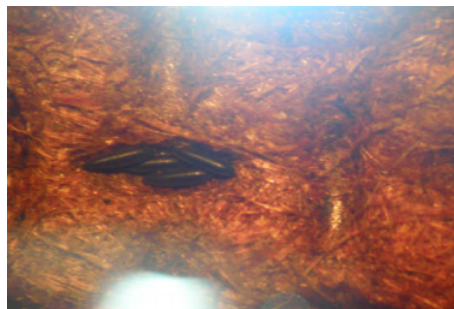
Fonte: (BAÊTA, 2007).

Figura 4 – Técnico fazendo a contagem de ovos com auxílio de um estereomicroscópio. Esta contagem é feita de forma manual.



Fonte: (BAÊTA, 2007).

Figura 5 – Imagem de microscopia de uma palheta de ovitampa com presença de ovos.



Fonte: (BAÊTA, 2007).

Uma das formas que podem contribuir para uma maior celeridade no processo de contagem é a construção de um sistema de visão computacional, pois com um sistema é possível automatizar o processo de digitalização das palhetas e a contagem, além de poder, armazenar informações referentes as palhetas. A próxima seção explica o que é um sistema de visão computacional e os seus componentes.

2.2 SISTEMA DE VISÃO COMPUTACIONAL

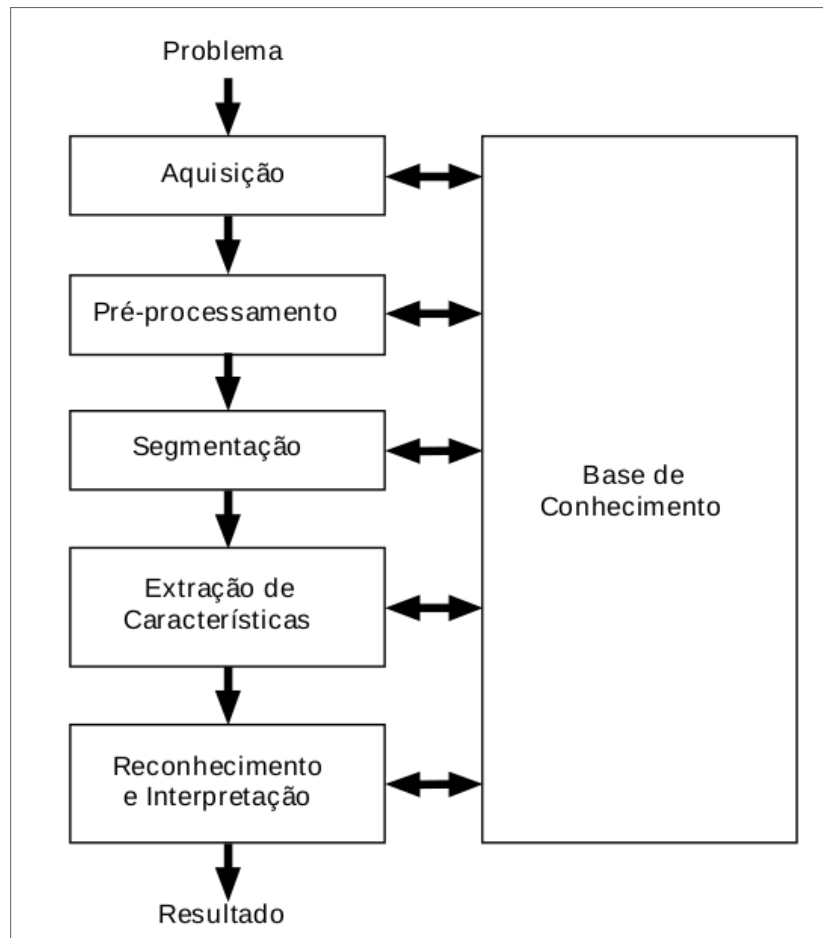
Visão computacional é utilizado para realizar análise digital em imagens para extrair informações relevantes, tornando possível tirar conclusões acerca da imagem como, por exemplo, saber qual é a cor presente na imagem ou até informações mais complexas, como reconhecer de quem é o rosto que aparece numa determinada cena (PARKER, 2010).

Segundo Gonzalez e Woods (2010), uma imagem pode ser definida como uma função bidimensional, $f(x, y)$, em que x e y são as coordenadas do plano cartesiano, e a amplitude de f em qualquer par de coordenadas (x, y) é chamada de intensidade ou nível de cinza da imagem nesse ponto. Quando x , y e os valores de intensidade de f são quantidades finitas e discretas, se tem uma imagem digital.

O Processamento Digital de Imagem (PDI) é frequentemente usado para aprimorar uma imagem para processamento posterior por algoritmos, além de possibilitar a manipulação e interpretação de imagens por meio das características digitais nelas contidas (PARKER, 2010). A partir das técnicas de PDI e de algoritmos de visão computacional, é possível criar sistemas de visão computacional.

De acordo com Filho e Neto (1999) a estrutura de um sistema de visão artificial ou sistema de visão computacional é composta pelos seguintes passos: aquisição da imagem, pré-processamento, segmentação, extração de características, reconhecimento e interpretação. O fluxo de funcionamento do sistema é mostrado na figura 6.

Figura 6 – Passos fundamentais de processamento em um sistema de visão computacional.



Fonte: (FILHO; NETO, 1999)

Filho e Neto (1999) definem cada uma das etapas apresentadas na Figura acima:

- a) Aquisição: é o primeiro passo, é nela que as imagens são capturadas, para cada projeto são consideradas particularidades, desta forma, é uma etapa importante para escolher qual é a melhor câmera para captura as imagens.
- b) Pré-processamento: a função da etapa de pré-processamento é a aplicação de filtros específicos como realce de contrastes, remoção de ruídos, entre outros, para melhorar os aspectos da imagem obtida, pois nem sempre as imagens obtidas estão perfeitas, com o pré-processamento adequado as imagens ficam melhores para as etapas seguintes.
- c) Segmentação: o objetivo desta etapa é dividir uma imagem em suas unidades significativas, ou seja, nos objetos de interesse que a compõem. É uma das etapas mais complexas de implementar.

- d) Extração de Características: esta etapa procura extrair características das imagens resultantes da segmentação por meio de descritores que permitam caracterizar bem estes objetos, nesta fase a entrada é uma imagem, mas a saída é um conjunto de dados correspondentes àquela imagem.
- e) Reconhecimento e Interpretação: nesta última etapa do sistema, denominamos reconhecimento o processo de atribuição de um rótulo a um objeto baseado em suas características, traduzidas por seus descritores. A atribuição de rótulos é o reconhecimento. Atribuir um significado ao objeto reconhecido trata-se de interpretação.
- f) Base de Conhecimento: todas as tarefas das etapas descritas acima pressupõem a existência de um conhecimento sobre o problema a ser resolvido, armazenado em uma base de dados, cujo tamanho e complexidade podem variar enormemente.

Cada etapa do processo de PDI pode ser melhorada e automatizada com o auxílio de Inteligência Artificial. O próximo tópico dissertação aborda sobre Inteligência Artificial e aprendizagem de máquina.

2.3 INTELIGÊNCIA ARTIFICIAL E APRENDIZADO DE MÁQUINA

Inteligência Artificial (IA) trata-se de um campo da ciência e engenharia capaz de construir máquinas inteligentes, focando em programas de computador (MCCARTHY, 1998). Esses programas buscam agir como humanos para solucionar problemas, a questão de agir como um humano surge principalmente quando os programas de IA precisam interagir com as pessoas, explicando seus resultados como, por exemplo, um sistema de diagnóstico clínico, ou um sistema que dialoga com usuário por meio de um *chat*. Esses programas devem se comportar de acordo com certas convenções normais de interação humana para serem compreendidos. A representação e o raciocínio nesses sistemas podem ser ou não baseados em um modelo humano (RUSSELL; NORVIG, 2002).

O campo de pesquisa em IA começou após a Segunda Guerra Mundial, quando pessoas começaram a trabalhar independentemente no desenvolvimento de máquinas inteligentes. Acredita-se que Alan Turing possa ter sido o pioneiro no desenvolvimento de máquinas inteligentes, visto que o referido cientista matemático mesmo deu uma palestra sobre IA em 1947, além disso, ele também pode ter sido essencial na decisão de que IA seria melhor através da programação de computadores e não pela construção de máquinas.

Atualmente é possível observar a aplicação da IA na resolução de problemas nos mais diversos setores econômicos, bem como no desenvolvimento de pesquisas científicas e tecnológicas que podem trazer benefícios à sociedade. Como exemplo, a presente pesquisa, que utiliza a Inteligência Artificial para contagem de ovos do mosquito *Ae. aegypti*, com o objetivo de obter um melhor monitoramento dessa população, visando uma melhor tomada de decisão quanto às medidas de prevenção para poderem minimizar a propagação de doenças transmitidas por este vetor, trazendo assim um benefício para autoridades de saúde pública, bem como para a sociedade em geral.

Um dos campos da IA que permite tais avanços é o de Aprendizagem de Máquina (AM), do termo original em inglês *Machine Learning*, é um campo de pesquisa da Inteligência Artificial que possui como objetivo desenvolver programas de computador que possam aprender e executar uma tarefa pré-determinada por meio da sua própria experiência (CARVALHO et al., 2011). Mitchell (1997) define a AM da seguinte forma:

“Um programa de computador é dito aprender a partir de uma experiência E com respeito a uma classe de tarefas T e medida do desempenho P, se seu desempenho nas tarefas em T, segundo a medida P, melhora com a experiência E”

O campo de pesquisa em AM está em constante crescimento e nele são estudados métodos, técnicas e ferramentas computacionais, que visam adquirir novos conhecimentos e novas habilidades capazes de melhorar o desempenho de algoritmos através da experiência (MITCHELL, 1997). Sendo assim, é possível afirmar que os programas desenvolvidos com a utilização de AM conseguem aprender sozinhos, por meio da utilização de um conjunto de dados que refletem experiências passadas.

Os sistemas de aprendizado possuem características únicas, possibilitando a classificação de acordo com seu modo, paradigma e tarefas de aprendizado. De acordo com Goldschmidt (2010) os principais algoritmos de aprendizado existentes baseiam-se no aprendizado indutivo, onde é comum que a obtenção do modelo de conhecimento seja abstraída de uma ou várias iterações sobre um conjunto de exemplos históricos. O aprendizado indutivo pode ser subdividido em aprendizado supervisionado e aprendizado não supervisionado, possuindo as seguintes definições:

Aprendizado supervisionado: neste tipo de aprendizado, é informado um conjunto de dados já rotulados em que já se conhece a saída correta e que deve ser semelhante ao conjunto, ou

seja, existe uma relação entre a entrada e a saída. A partir disso é realizada a análise do resultado esperado com o resultado produzido pelo modelo.

Aprendizado não supervisionado: nesta categoria de aprendizado, o conjunto de dados fornecido não é rotulado e a saída é previamente desconhecida. Possui como objetivo estabelecer a existência de grupos ou similaridades nestes conjuntos de dados.

Na tangente das tarefas de aprendizado, geralmente os problemas de aprendizado supervisionado são classificados como problemas de regressão e classificação. Enquanto os problemas de classificação buscam prever os resultados em uma saída discreta, os problemas de regressão buscam prever os resultados em uma saída contínua. Este trabalho explora técnicas de aprendizado de máquina, de maneira supervisionada e relacionadas especificamente às tarefas de classificação e detecção de modo a investigar estratégias para detecção e contagem automática de ovos do mosquito *Ae. aegypti*. A próxima seção deste trabalho abordará sobre Redes Neurais Artificiais um tópico que também vem evoluindo muito dentro da IA a partir da maior disponibilidade de informação e maior poder computacional das máquinas atualmente.

2.4 REDES NEURAIAS ARTIFICIAIS

A definição Rede Neural Artificiais (RNAs) apresentada por Braga Braga (2000), é que RNAs são como “sistemas paralelos distribuídos compostos por unidades de processamento simples (nodos) que calculam determinadas funções matemáticas”. As RNAs se caracterizam por ser uma computação não algorítmica que lembra a estrutura do cérebro humano. Haykin (2001) define uma rede neural como “uma máquina projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse”. Ademais, as redes neurais se assemelham com o cérebro humano em dois aspectos, de acordo com Haykin (2001)

“1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem. 2. Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido”.

Segundo Goldschmidt (2010), nas redes neurais artificiais as informações devem ser processadas tendo como princípio a organização de neurônios do cérebro, ou seja, assim como o cérebro consegue aprender e tomar decisões baseadas em aprendizagem, as redes neurais devem buscar fazer o mesmo. Portanto, uma RNAs deve conseguir armazenar conhecimento

baseado em experiências e posteriormente disponibilizar o conhecimento para uma determinada aplicação.

Goldschmidt (2010) afirma que como as RNAs se assemelham com a estrutura do cérebro, elas também possuem algumas características similares ao comportamento humano, dentre elas:

- a) Busca paralela e endereçamento pelo conteúdo: o cérebro não possui endereço de memória e não procura a informação de maneira sequencial. Semelhantemente, nas RNAs o conhecimento fica distribuído na estrutura das redes, então, a busca pela informação ocorre de maneira paralela e não sequencial.
- b) Aprendizado por meio da experiência: as redes neurais buscam aprender padrões por meio de dados inseridos. Isto ocorre a partir da utilização de um processo que apresenta repetidamente os dados à rede, que no que lhe concerne, busca abstrair modelos de aprendizado de forma automática. Este processo se chama aprendizado sendo realizado por meio dos algoritmos de aprendizado.
- c) Generalização: trata-se de uma das principais características das RNAs, estas conseguem generalizar seu conhecimento a partir de exemplos anteriores, ou seja, as RNAs podem aprender a partir de um conjunto reduzido de exemplos e dar respostas corretas para os dados que não são conhecidos. Esta capacidade também permite que as RNAs consigam responder corretamente a novos padrões mesmo que os dados possuam ruídos e distorções.
- d) Associação: esta característica refere-se à capacidade da rede fazer relações entre diferentes padrões. Como exemplo, a identificação de pessoas através de suas características de voz.
- e) Abstração: trata-se da capacidade de abstrair a essência de um conjunto de dados de entrada. As RNAs conseguem perceber quais as características relevantes em um conjunto de entradas, portanto, elas conseguem, a partir de padrões ruidosos, extrair informações dos padrões sem ruído.
- f) Robustez e Degradação Gradual: esta característica afirma que a perda de um conjunto de neurônios artificiais não causa o mau funcionamento da rede neural.

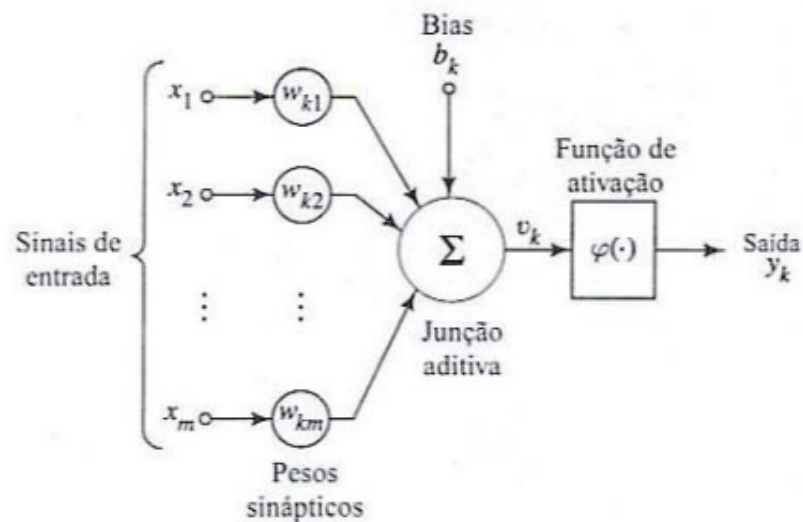
g) Soluções aproximadas: nem sempre as RNAs produzem a melhor solução para um problema, no entanto, geram soluções aproximadas e aceitáveis.

As redes neurais se formam por meio dos neurônios e conexões entre eles. O neurônio artificial representa uma região cujas informações são processadas. Um modelo não-linear de um neurônio é representado conforme a é formado por três elementos básicos (HAYKIN, 2001):

- Um conjunto de sinapses ou elos de conexão (x_1, x_2, \dots, x_n), caracterizados por um peso ou força ($w_{k1}, w_{k2}, \dots, w_{kn}$);
- Um somador (Σ) para acumular os sinais de entrada;
- Uma função de ativação ($\varphi(\cdot)$) que restringe o intervalo permissível de amplitude do sinal de saída (y) a um valor finito;

A Figura 7 mostra essa configuração de m modelo não-linear.

Figura 7 – Modelo não-linear de um neurônio .



Fonte: (HAYKIN, 2001)

O modelo neuronal também apresenta um bias que é aplicado externamente, representado por b_k e possui efeito para aumentar ou diminuir a ação da função de ativação. Em termos matemáticos, o neurônio pode ser descrito por meio das seguintes equações:

$$v_k = \sum_{j=1}^n w_{kj} \cdot x_j \quad (2.1)$$

$$y_k = \varphi(w_k + b_k) \quad (2.2)$$

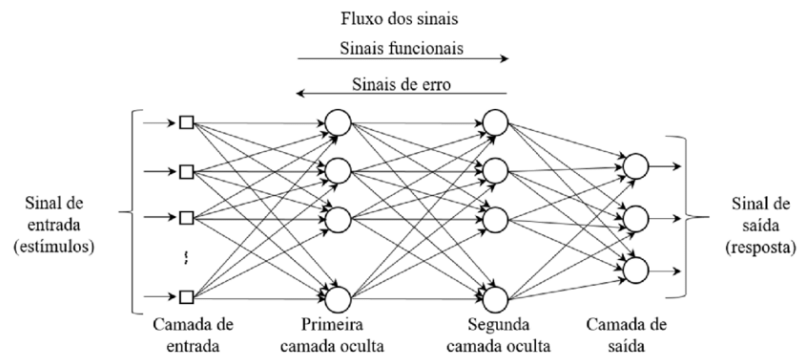
Onde:

- x_1, x_2, \dots, x_n são os sinais de entrada do neurônio k
- $w_{k1}, w_{k2}, \dots, w_{kn}$ são os pesos do neurônio k
- u_k é o resultado do somador
- b_k é o bias
- $\varphi(\cdot)$ é a função de ativação
- y_k é o sinal de saída do neurônio k

De acordo com Haykin (2001), para se construir uma RNAs completa é preciso definir uma topologia que representará sua estrutura, em relação à distribuição dos neurônios, definição de camadas e suas conexões. A literatura apresenta algumas topologias tradicionais, tais como: Perceptron de Única Camada, Adaline, Madaline, Perceptron de Múltiplas Camadas, dentre outras.

No modelo Perceptron Multicamadas, do inglês *Multilayer Perceptron* (MLP), as redes neurais consistem em múltiplas camadas de nós computacionais. Haykin (2001) afirma que no modelo MLP a rede consiste em um conjunto de unidades sensoriais que constituem a camada de entrada, uma ou mais camadas ocultas de nós computacionais e uma camada de saída de nós computacionais. Esta topologia apresenta processamento *feedforward*, ou seja, cada neurônio existente na camada tem conexões com todos os neurônios da camada seguinte.

Figura 8 – Modelo Perceptron Multicamadas.



Fonte: (GOODFELLOW et al., 2016)

Como pontua Haykin (2001), uma rede MLP possui três características distintivas:

1. Cada neurônio da rede possui uma função de ativação não linear.
2. A rede possui uma ou mais camadas de neurônios ocultos, que não fazem parte nem da entrada, nem da saída da rede. Estes neurônios permitem que a rede possa aprender tarefas complexas através de extração das características mais importantes dos padrões de entrada.
3. A rede possui um alto grau de conectividade, sendo determinado com a ajuda das sinapses da rede.

Segundo Goldschmidt (2010) é extremamente importante que seja definido o número de neurônios em cada camada da rede MLP, pois afeta o desempenho da rede e sua capacidade de generalização, e, embora não haja regras para definição precisa do número, existem algumas heurísticas que auxiliam na escolha do número de camadas, como, por exemplo, a de Hecht-Nielson, que sugere a utilização de $(2n+1)$ neurônios na camada intermediária, sendo N o número de neurônios existentes na camada de entrada. Além da definição do número de neurônios em cada camada da rede, de acordo com Haykin (2001), para que uma rede MLP funcione adequadamente é necessário que seja realizado o treinamento supervisionado por meio de um algoritmo de aprendizado, pois permite que a rede neural aprenda com um conjunto de dados conhecidos e consiga generalizar as respostas para entradas desconhecidas.

As redes MLPs possuem diversos algoritmos de treinamento, no entanto, o mais popular é o *Back-Propagation* (BP), presente em inúmeros casos de sucesso e capaz de realizar o treinamento de uma RNAs, auxiliando na resolução de problemas complexos. O algoritmo

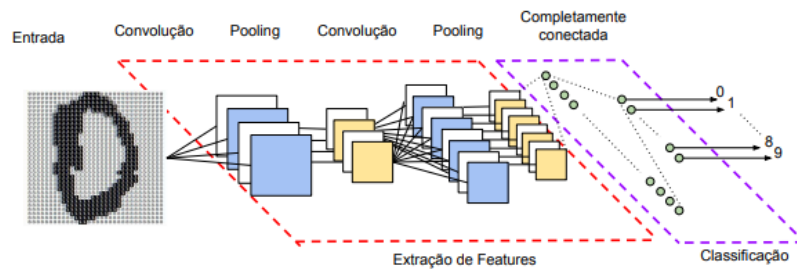
BP é baseado na regra de aprendizado por correção de erro, de modo que o treinamento é realizado através da inserção de um conjunto de dados na entrada e saída esperada. De acordo com Goldschmidt o BP utiliza pares de entrada e saída para, por meio de correção de erros, ajustar os pesos da rede.

A partir do modelo de MLP, as RNAs foram evoluindo com o surgimento de diversas abordagens e modificações na estrutura e na forma como elas aprendem e transmitem o conhecimento em suas arquiteturas, hoje há diversas áreas sendo beneficiadas com aplicações que recorrem a de RNAs, por exemplo, problemas como carros autônomos, reconhecimento de voz, corretores de texto, classificação e detecção de objetos em imagens, aplicação em diagnósticos médicos, entre outros. Entre as áreas que mais evoluíram está a visão computacional com Redes Neural Convolutiva, do inglês *Convolutional Neural Network (CNN)*, que será explanada no tópico seguinte.

2.4.1 Redes Neurais Profundas Convolucionais

Uma CNN é uma RNAs baseada na topologia MLP, possui inspiração no processamento biológico de processamento de dados visuais. Segundo Vargas, Paes e Vasconcelos (2016), a CNN consegue aplicar filtros em dados visuais e manter uma relação de vizinhança entre os píxeis da imagem ao longo do processamento da rede. Esta categoria de RNAs está sendo bastante utilizada nas aplicações que realizam classificação, detecção e reconhecimento em imagens e vídeos. Uma CNN, assim como a MLP, é composta por uma camada de entrada, várias camadas intermediárias e uma camada de saída. As camadas intermediárias desta rede podem ser divididas em dois grupos: extração de características e classificação. O termo profunda está relacionado essencialmente as RNAs com três ou mais camadas, visto que, essas redes podem chegar a centena de camadas.

Figura 9 – Exemplo de uma rede neural convolucional. Na Figura é possível observar uma imagem do dígito zero entrando numa CNN que a faz o processo convolução, *pooling* e classificação de qual dígito é a imagem, neste caso, é esperado que a CNN classifique como dígito zero.



Fonte: (VARGAS; PAES; VASCONCELOS, 2016)

A Figura 9 demonstra um exemplo de CNN. Inicialmente, é apresentada a camada de entrada, representada por uma imagem. Após a entrada, fica a camada de convolução, que possui os mapas de características gerados a partir dos filtros aprendidos. Em cada filtro convolucional estão presentes vários neurônios clonados, que visualizam características diferentes da imagem de entrada. Na sequência, está a camada de *pooling*, seguida por outra camada de convolução e mais uma de *pooling*, que compõem o grupo de extração de características. O segundo grupo conceitual apresenta duas camadas completamente conectadas, responsáveis por realizar a classificação final da rede.

De acordo com Vargas, Paes e Vasconcelos (2016) a camada de convolução é composta por vários neurônios, e cada um deve aplicar um filtro em um pedaço específico da imagem. Já a camada de *pooling* possui como objetivo reduzir a dimensionalidade dos dados nas redes, buscando tanto agilidade no treinamento, quanto a criação de uma invariância espacial.

As CNNs foram popularizadas a partir do ano de 2012 com a rede AlexNet que ganhou a competição científica *ImageNet Large Scale Visual Recognition (ILSVRC)* que é o desafio de reconhecimento visual de grande escala da ImageNet. Esta competição consiste em avaliar diversos algoritmos para detecção de objetos e classificação de imagens em larga escala. Até então, os algoritmos que trabalhavam com detecção e classificação em imagens eram dependentes de engenharia de atributos, ou seja, eram analisados padrões por especialistas e esses padrões eram decodificados para serem encontrados nas imagens. AlexNet é uma rede que consegue sozinha aprender a extrair as características necessárias das imagens e, a partir delas, realizar a classificação/regressão/agrupamento das informações de entrada (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). O crescimento da disponibilização de informação e o aumento no poder de processamento das máquinas com unidade gráfica de processamento, do inglês

Graphics Processing Unit (GPU) também contribuíram para que AlexNet conseguisse realizar um trabalho tão inovador.

Desde então, foram lançadas para a mesma competição, redes como a ResNet da Microsoft que obteve um erro de 3,6, conseguindo ter um erro menor que o de humanos que são de 5,0. Algo bastante positivo desta competição é que as redes treinadas nestas bases de dados com milhões de imagens são disponibilizadas publicamente e pesquisadores do mundo inteiro podem fazer adaptações e treiná-las em seus conjuntos de dados específicos, pois é possível reaproveitar o aprendizado destes modelos em outras tarefas utilizando o conceito de Transferência de Aprendizado (do inglês *Transfer Learning*).

2.4.2 EfficientDet

Efficient Object Detection (EfficientDet) é uma CNN lançada pela equipe do Google que propõe uma melhoria na CNN EfficientNet. As principais contribuições da EfficientDet (TAN; PANG; LE, 2020):

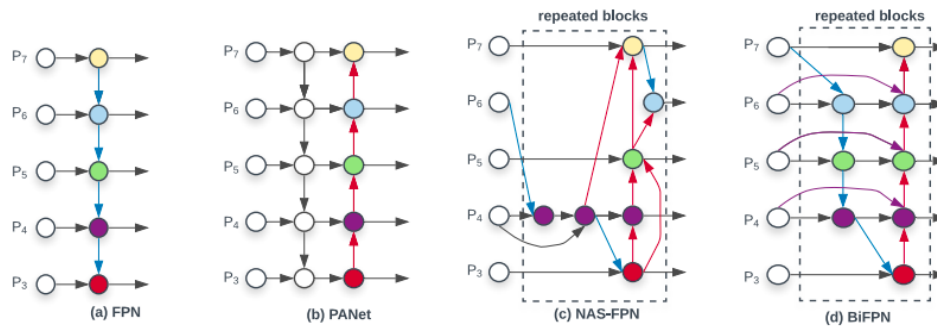
- a) Weighted Bi-directional Feature Pyramid Network (BiFPN), do inglês *Weighted Bi-directional Feature Pyramid Network*, permite fusão rápida de recursos em múltiplas escalas bidirecionais.
- b) Um novo método de dimensionamento composto em conjunto dimensiona *backbone*, rede de recursos, rede de caixa/classe e resolução, entende-se como classe o objeto alvo na imagem.

O objetivo do BiFPN é agregar recursos em várias escalas de maneira eficaz de cima para baixo. A fusão de recursos em várias escalas visa agregar recursos em resoluções diferentes. Formalmente, dada uma lista de multi-escala-recursos $\vec{P}^{in} = (P_{l_1}^{in}, P_{l_2}^{in}, \dots)$ onde $P_{l_i}^{in}$ representa o recurso no nível l_i , o objetivo é encontrar uma transformação f que possa efetivamente agregar diferentes recursos e gerar uma lista de novos recursos: $\vec{P}^{out} = f(\vec{P}^{in})$ (TAN; PANG; LE, 2020).

A figura 10 mostra a comparação de diferentes métodos de agregação baseados em Feature Pyramid Network (FPN). Começando em (a), uma rede FPN típica funde recursos de várias escalas do nível 3 ao 7 (P3 - P7). A arquitetura PANet (b) adiciona um caminho ascendente adicional na parte superior do FPN para usar os recursos de nível inferior de forma eficiente.

A conexão entre os nós também podem ser encontradas usando uma pesquisa de arquitetura neural. Conforme mostrado em (c), as conexões diferem das construídas por humanos. Em (d) é BiFPN com melhor precisão e compensações de eficiência (TAN; PANG; LE, 2020).

Figura 10 – Diferentes métodos de agregação baseados em FPN. A principais diferenças são desempenho e a eficiência. A PANet atinge melhor precisão do que a FPN e a NAS-FPN, entretanto tem mais parâmetros. Já BiFPN otimiza esses modelos com conexões de escala cruzada.



Fonte: (TAN; PANG; LE, 2020)

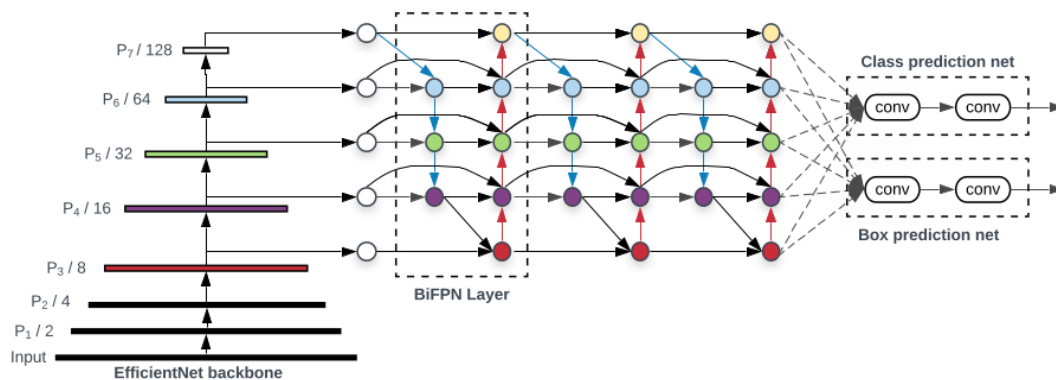
A Figura 10 mostra em (a) o FPN convencional de cima para baixo. Leva recursos de entrada de nível 3-7 $\vec{P}^{in} = (P_3^{in}, \dots, P_7^{in})$, em que P_i^{in} representa um nível de recurso com resolução de $1/2^i$ das imagens de entrada. Por exemplo, se a resolução de entrada for 640x640, então P_3^{in} representa o nível de recurso 3 ($640/2^3 = 80$) com resolução 80x80, enquanto P_7^{in} representa o nível de recurso com resolução 5x5. O FPN convencional agrega escala múltipla recursos de cima para baixo. Entretanto, a FPN convencional de cima para baixo é inerentemente limitado pelo fluxo de informação unilateral. Para resolver este problema, PANet adiciona uma rede extra de agregação de caminho de baixo para cima, como mostrado na 10 (b), a NAS-FPN emprega pesquisa de arquitetura neural para procurar melhor escala cruzada para a topologia da rede, mas requer muitas horas de GPU durante a pesquisa e a rede encontrada é irregular e difícil de interpretar ou modificar, como mostra a 10 (c) (TAN; PANG; LE, 2020).

Ao estudar o desempenho e a eficiência dessas três redes, foi observado que a PANet atinge melhor precisão do que a FPN e a NAS-FPN, mas com o custo de mais parâmetros e cálculos. Para melhorar a eficiência do modelo, os autores propõem várias otimizações para conexões de escala cruzada em três situações: primeiro, se um nó tem apenas uma borda de entrada sem fusão de recursos, então terá menos contribuição para rede que visa a fusão de recursos diferentes. Isso leva a uma rede bidirecional simplificada; segundo, é adicionada uma borda extra na entrada original para o nó de saída, se eles estiverem no mesmo nível, junta

mais recursos sem adicionar muito custo; terceiro, ao contrário do PANet que tem apenas um top-down e um caminho de baixo para cima, aqui cada bidirecional (de cima para baixo e de baixo para cima) caminha como uma camada de rede de recurso e repete a mesma camada várias vezes para permitir mais fusão de recursos de alto nível. Com essas otimizações, se tem a nova rede de recursos como rede em pirâmide de recursos bidirecionais (BiFPN) (TAN; PANG; LE, 2020).

A Figura 11 mostra a arquitetura geral da EfficientDet, os autores empregaram EfficientNets pré-treinadas no ImageNet como rede de *backbone*. O BiFPN proposto serve como a rede de recursos, que obtém recursos de nível 3-7 P3, P4, P5, P6, P7 da rede de *backbone* e aplica repetidamente a fusão de recursos bidirecionais de cima para baixo e de baixo para cima. Esses recursos combinados são fornecidos a uma rede para produzir classes de objeto e previsões em caixa delimitadora. Os pesos da rede de caixa delimitadora são compartilhados em todos os níveis de recursos.

Figura 11 – Arquitetura geral da EfficientDet. Utilizam redes pré-treinadas no ImageNet como rede de *backbone*. BiFPN Funciona como extrator de recursos, por fim, os recursos são combinados e utilizados pela rede para fazer a predição de classes.



Fonte: (VARGAS; PAES; VASCONCELOS, 2016)

A EfficientDet foi avaliada no conjunto de dados de detecção COCO 2017 com imagens de treinamento de 118K. Cada modelo é treinado usando o otimizador SGD com momentum 0,9 e queda de peso $4e-5$. A taxa de aprendizagem é linearmente aumentada de 0 a 0,16. Começa no modelo EfficientDet0 até o modelo EfficientDet-D7. A rede EfficientDet-D7 atinge o estado da arte de 55,1 AP no teste-dev COCO com parâmetros 77M e 410B FLOPs1, sendo 4x - 9x menor e usando 13x - 42x menos FLOPs do que os detectores anteriores.

2.4.3 YOLOv4

YOLO, do inglês *You Only Look Once*, é uma CNN para detecção de objetos em imagens, ela divide uma imagem em regiões e, em seguida, prevê as caixas delimitadoras e suas probabilidades para cada região, ela faz isso com apenas uma execução do algoritmo, pois trabalha com regressão, diferentemente de outros modelos que funcionam como classificação. Isso torna a YOLO extremamente rápida, em sua versão mais básica, ela funciona a 45 quadros por segundo sem processamento em lote.

Neste projeto, utilizamos a versão mais nova da YOLO que é versão 4. Comparada às versões anteriores, a YOLOv4 melhorou no processamento de dados. O *backbone*, o treinamento da rede, a função de ativação e a função de perda foram otimizados, o que tornou a versão mais rápida fazendo alcançar o melhor equilíbrio entre precisão e velocidade nestes algoritmos de detecção de objetos em tempo real (BOCHKOVSKIY; WANG; LIAO, 2020).

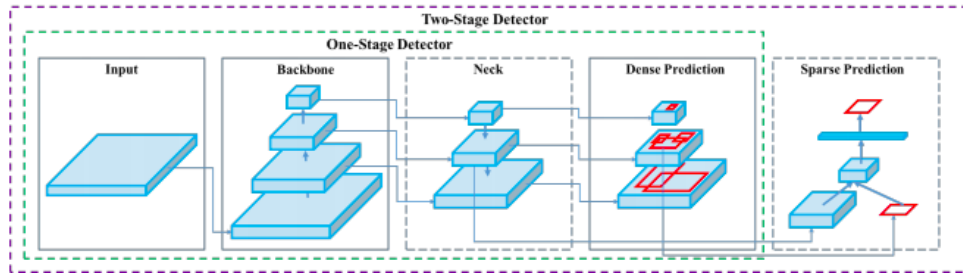
Segundo Bochkovskiy, Wang e Liao (2020), as principais contribuições da YOLOv4 são:

- a) É um modelo de detecção de objetos eficiente e poderoso que pode ser treinado em GPUs 1080 Ti ou 2080 Ti.
- b) Verificaram a influência dos métodos de detecção de objetos *Bag-of-Freebies* e *Bag-of-Specials* de última geração durante o treinamento do detector. *Bag-of-Freebies* é um conjunto de técnicas ou métodos que mudam a estratégia de treinamento ou o custo do treinamento para melhorar a precisão do modelo. Alguns dos métodos são: aumento de dados, polarização de distribuição semântica em conjuntos de dados, e Função Objetivo da Regressão BBox. Enquanto *Bag-of-Specials* contém diferentes *plugins* e módulos de pós-processamento que apenas aumentam o custo de inferência em uma pequena taxa, mas melhora drasticamente a precisão do detector de objetos.
- c) Modificaram os métodos de última geração como: CBN, PAN, SAM, tornando-os mais eficiente e adequado para treinamento em GPU.

Uma rede de detecção moderna é composta por duas partes, um backbone que é pré-treinado na ImageNet e um head usado para prever classes e caixas delimitadoras de objetos como é possível ver na Figura 12. A arquitetura da YOLOv4 é composta de CSPDarknet53 como *backbone* que é uma estrutura de rede neural de código aberto para treinar e extrair recursos de imagem, um módulo adicional de *pooling* de pirâmide espacial que foi empregado

como a rede de *neck* para conseguir uma melhor fusão das características extraídas, e, por fim, a YOLOv3 como head (BOCHKOVSKIY; WANG; LIAO, 2020).

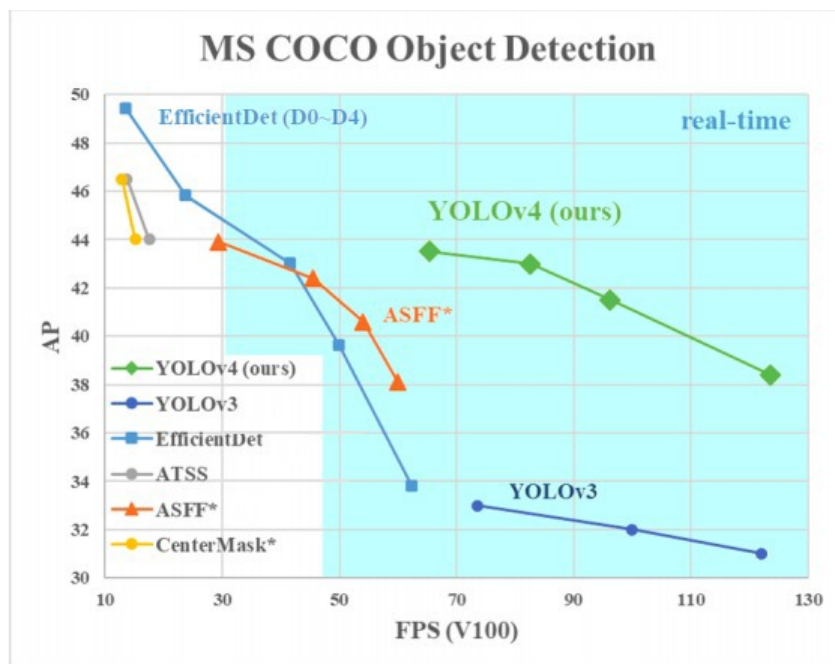
Figura 12 – Arquitetura da YOLOv4. O *input* é a imagem passada para a rede, em seguida, essa imagem entra no *backbone* que funciona como extrator de características, a parte de *neck* é empregada para fazer a fusão das melhores características. Por fim, nos dois últimos passos são efetuadas as predições.



Fonte: (BOCHKOVSKIY; WANG; LIAO, 2020)

A Figura 13 mostra a comparação da YOLOv4 proposto e outros modelos detectores de objetos de última geração. A YOLOv4 funciona duas vezes mais rápido do que EfficientDet com desempenho comparável. Melhora AP e FPS do YOLOv3 em 10% e 12%, respectivamente, como mostra a figura a seguir.

Figura 13 – Resultados da YOLOv4 na competição MS COCO comparada as redes YOLOv3, EfficientDet, ATSS e CenterMask.



Fonte: (BOCHKOVSKIY; WANG; LIAO, 2020)

Como é possível observar na Figura acima, a YOLOv4 tem tanto um bom resultado em FPS como em AP ao comparado entre as demais redes.

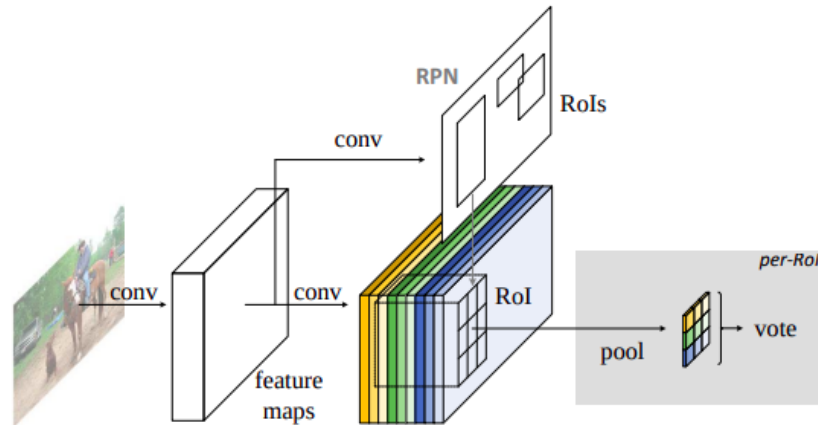
2.4.4 Region-based Fully Convolutional Network (RFCN)

A Region-based Fully Convolutional Network (RFCN) é um modelo de rede neural para detecção de objetos em imagens que busca melhorar Fast/Faster e RCNN que também são duas redes detectoras baseadas em Rols (Regiões de Interesse), mas que aplicam uma sub-rede custosa centenas de vezes em cada ROI, enquanto a RFCN é uma rede totalmente conectada baseada em ROI que compartilha quase toda sua computação na imagem. Com isso, A RFCN melhora a velocidade reduzindo a quantidade de trabalho necessária para cada ROI. Os mapas de recursos baseados em regiões são independentes dos Rols e podem ser calculados fora de cada ROI. Esse processo torna a RFCN mais rápida do que a Fast R-CNN e a Faster R-CNN (DAI et al., 2016).

A RFCN consiste em arquiteturas compartilhadas e totalmente convolucionais, como é o caso do FCN. Para resolver o dilema de *translation/variance* na classificação e detecção de objetos em imagens da FCN, foi construído um conjunto de mapas de pontuação sensíveis à posição usando um banco de camadas convolucionais especializadas como a saída da FCN. Cada um desses mapas de pontuação codifica as informações de posição em relação a uma posição espacial relativa. Além da FCN, foi acrescentada uma camada de *pooling* de Rol sensível à posição que orienta as informações desses mapas de pontuação, sem seguir camadas de peso (convolucional/fc). Toda a arquitetura é aprendida de ponta a ponta. Todas as camadas que podem ser aprendidas são convolucionais e compartilhadas em toda a imagem, mas codifica as informações espaciais necessárias para a detecção de objetos (DAI et al., 2016).

A RFCN usa a rede residual de 101 camadas (ResNet-101) como backbone, esta configuração obteve 83,6% de mAP no conjunto PASCAL VOC 2007 e 82,0% no conjunto 2012. além de ter obtido boas taxas de velocidade inferência no conjunto de teste com 170ms por imagem que é $2,5\times$ a $20\times$ mais rápido do que a Faster R-CNN com backbone ResNet-101 (DAI et al., 2016). Sua arquitetura pode ser vista na Figura 14.

Figura 14 – Arquitetura do modelo de rede neural RFCN para detecção de objetos. Da esquerda para a direita: O modelo recebe uma imagem como entrada e aplica seu processamento para identificar o objeto.



Fonte: (DAI et al., 2016)

A figura 14 mostra a arquitetura geral da RFCN que é uma Rede de Proposta de Região (do inglês Region Proposal Network - RPN) que propõe Rols candidatos, que são aplicados nos mapas de pontuação. Todas as camadas de pesos aprendidos são convolucionais e calculadas em toda a imagem; o custo computacional por Rol é insignificante.

Dado os Rols, A RFCN é projetada para classificar os Rols em categorias de objeto e plano de fundo. Na RFCN, todas as camadas de peso aprendidas são convolucionais e calculadas em toda a imagem. k^2 mapas de pontuação sensíveis à posição para cada categoria e, tem $k^2(C+1)$ - camada de canal de saída com C categorias de objeto (+1 para o fundo). O banco de mapas de pontuação k^2 corresponde a uma grade espacial $k \times k$ que descreve as posições relativas. Por exemplo, com $k \times k = 3 \times 3$, os 9 mapas de pontuação codificam os casos superior esquerdo, superior central, superior direito, ..., inferior direito de uma categoria de objeto. RFCN termina com uma camada de pooling Rol sensível à posição. Esta camada agrega as saídas da última camada convolucional e gera pontuações para cada Rol. Ao contrário de da Fast/Faster R-CNN, o Rol sensível à posição da camada conduz o pooling seletivo, e cada um dos $k \times k$ bin agrega respostas de apenas um mapa de pontuação fora do banco de mapas $k \times k$. Com treinamento de ponta a ponta, esta camada Rol orienta a última camada convolucional para aprender mapas de pontuação sensíveis à posição especializados (DAI et al., 2016).

Todos os modelos de CNNs citados acima foram adaptados para este trabalho para que fosse possível fazer uma comparação entre eles e saber qual deles melhor se encaixa no contexto de detecção e contagens de ovos de *Ae. aegypti*. A transferência de Aprendizado é útil, pois

treinar uma CNN do zero é um processo custoso computacionalmente, e a depender dos recursos de *hardware*, este treinamento pode durar dias. Portanto, A técnica de Transferência de Aprendizado é utilizada com o objetivo de reusar parte do “aprendizado” já adquirido por um modelo de rede neural em algum conjunto de dados, objetivando resolver outras tarefas, reduzindo assim, o tempo necessário para treinar um novo modelo (CARVALHO, 2015). Ao aplicar esta técnica em outras tarefas, um modelo base é utilizado na inicialização dos pesos ou como extrator de características primárias. A utilização desta técnica possibilitou a resolução de diversos problemas envolvendo visão computacional, pois além de diminuir o tempo de treinamento de novos modelos, também possibilitou treinar modelos em conjuntos de dados reduzidos, visto que, o processo de aquisição de dados pode ser uma tarefa complexa e custosa. A seguir serão apresentadas os modelos de rede neurais utilizados neste trabalho para fazer o processo de transferência de aprendizado.

2.5 CONCLUSÃO DO CAPÍTULO

O capítulo 2 apresentou uma revisão dos fundamentos gerais utilizados no desenvolvimento deste trabalho. Primeiramente foi apresentado sobre o *Ae. aegypti*, as transmissões de doenças causada por ele e os métodos de vigilância deste vetor. Na sequência foi apresentado a arquitetura de um sistema de visão computacional, objetivando dá um contexto de como construir uma ferramenta que soluciona problemas com imagens. Depois foi apresentado os tópicos de IA numa visão geral até entrar em RNAs mostrando mais profundamente as CNNs que são modelos de RNAs que trabalham com imagens e vem sendo populares em resolução de problemas nesta área.

3 TRABALHOS RELACIONADOS

Nesta seção serão apresentados os trabalhos relacionados a contagem automática de ovos de *Ae. aegypti*.

Gusmão, Machado e Rodrigues (2009) propõem um método para contar automaticamente o número de ovos em imagens digitais de ovitrampas com base em técnicas de processamento de imagem (exploração de sistemas de cores) e algoritmo de agrupamento k-Means. Para o experimento, os autores definiram o número de clusters para três. Tais clusters estão associados às seguintes classes: ovo, armadilha e regiões intermediárias. O algoritmo proposto por eles, seleciona aleatoriamente a média inicial dos vetores. O principal objetivo é minimizar a soma das distâncias euclidianas de cada objeto até a média do cluster vetor (centróide) a que estão associados, para cada cluster o conjunto total de clusters. Essa medida é chamada de coesão. Como Critério de parada, foram utilizadas duas medidas: uma máxima de iterações de 100; e a estimativa da coesão. O algoritmo pára quando a soma das distâncias para cada objeto não pode ser diminuída ainda mais. O algoritmo repete o agrupando 3 vezes, cada um com um novo conjunto de médias iniciais do cluster vetores posiciona e retorna a solução com a menor valor para as somas das distâncias dos vetores ponto a média.

Após agrupar as imagens, é necessário definir quais *clusters* contém ovos. Para esse fim, o RGB original imagem é convertida em HSV (H = matiz, S = saturação, V = valor) imagem. Matiz está relacionado com os tons que a imagem contém. Assim, uma análise do componente matiz fornece informações sobre os objetos. Um algoritmo de componentes conectados é aplicado a imagem nivelada para rotular suas regiões conectadas, este algoritmo coloca um rótulo diferente em cada área branca conectada da imagem. Com esta rotulagem, é possível medir cada área conectada. Pequenas áreas são excluídas, pois não poderiam conter um ovo. Por experimentações, foi definido que todas as áreas com menos de 140 pixels conectados em branco devem ser apagadas. Por fim, considerou-se que um ovo ocupa uma área média de 357 pixels. Assim, o número de ovos é a quantidade total de pixels brancos divididos por essa área média.

O método descrito atingiu um valor máximo erro individual de 45,45% em uma imagem em que há uma diferença de dez ovos. Mas, em geral, o erro médio foi 1,16% e o desvio padrão foi de cerca de 4,15%, o que é aceitável em comparação com um sistema não automático método. O erro médio é mais importante que os erros individuais, pois o método será aplicado

no imagem completa em uso prático.

Silva, Rodrigues e Araujo (2011) propõem a criação de um sistema e *hardware* que digitaliza a ovitrampas e permite que os usuários por meio de uma interface possa contar os ovos manualmente em imagens, ou o usuário permite que as imagens sejam enviadas para um servidor para que seja feita a contagem dos ovos automaticamente usando um *software* de reconhecimento de padrão, baseado em segmentação e filtragem.

O sistema é composto por uma unidade de controle, um dispositivo de captura de imagens, um sistema de iluminação de LED e um suporte mecânico associado a uma etapa de translação linear motorizada. A unidade de controle é baseada em um firmware/microcontrolador (PIC 16F876), que aciona o sistema mecânico e o dispositivo de captura de imagens, colocado 3 cm acima da palheta. O dispositivo de captura de imagens utilizado é uma câmera digital de 7,1 megapixels com modo de operação "super macro". O dispositivo de captura escolhido permite obter imagens com 3072 pixels x 2304 pixels e tamanho de arquivo de 3,5 MBytes no formato JPEG. O tamanho do arquivo gerado minimiza o custo computacional da análise automática da imagem e permite a identificação visual dos ovos de *Ae. aegypti*.

Os autores fazem a validação do sistema comparando os resultados da contagem automática com os obtidos pelo processo de contagem manual, eles verificaram que a contagem automática é mais rápida. Na avaliação de 100 ovitrampas, foram identificados 3297 ovos pelo procedimento manual no computador e 3385 na contagem automática.

Garcia et al. (2019) desenvolveram uma forma de aquisição de imagens que possa ser replicado em qualquer laboratório, com uma infraestrutura mínima, eles construíram uma base com 360 imagens, sendo 300 imagens para treinamento, 30 para validação e 30 para teste. As imagens são extraídas no formato RGB e tem dimensões de 3000 x 4000. Depois de coletar as imagens, é feito uma segmentação nas imagens para remover áreas pretas ao redor da região de interesse, pois uma parte da imagem não recebe iluminação. Após extrair a área de interesse, os autores, passam as imagens para um classificador de árvore de decisão para classificar cada pixel da imagem como pertencente ou não ao ovo. Depois do classificador incorporado, são geradas máscaras de imagens separando os pixels que supostamente pertencem ao ovo. A partir dessas máscaras e análises humanas, novos exemplos são gerados para alimentar a árvore.

Depois de construir a árvore, um refinamento é realizado para eliminar objetos que são menores que um tamanho mínimo por eles definido. Com o resultado gerado, é alimentada uma CNN pré-treinada em 10 classes. O objetivo da CNN é classificar se é ovo ou um fundo. Após isso, é treinado uma R-CNN (regiões com convoluções detector de objetos baseado em

redes neurais) para gerar as caixas delimitadoras dos ovos detectados. Por fim, é analisada a contagem do número de ovos com base no número de pixels, é possível correlacionar o número de ovos reais na imagem com o número extraído pelo modelo, através de uma regressão linear.

Os autores analisam os resultados da inferência em 30 imagens. Eles decidiram fazer duas análises, pois uma primeira análise teve um resultado não satisfatório, pois 3 imagens do conjunto apresentaram um erro maior que o normal. Uma imagem que contém 331 ovos, apenas 126 foram encontrados, outro com 232 foi encontrado 161 e, finalmente, uma imagem que não tem ovos foi encontrada em 77. Depois da retirada dessas imagens, o algoritmo conseguiu encontrar 91% dos ovos nas imagens, considerando uma taxa de interseção na união de 0,3. A correlação entre as contagens realizadas pela a CNN e a contagem real, onde C é 0,5468, R^2 tem o valor de 0,65 e R é próximo de 0,86, o que pode ser considerado uma forte correlação.

Brun et al. (2020) fazem um extenso levantamento dos principais trabalhos publicados que propõem o uso de computação automação da contagem de mosquitos da dengue.

Tabela 1 – Trabalhos Relacionados

Trabalho	Técnica	Validação	Principais Resultados
Mains, Mercer e Dobson (2008)	Quantização para tons de cinza Limiarização simples com um patamar análise de componentes conectados	Não Informado	R = 0,9945 A. aegypti R = 0,9925 A. albopictus R = 0,9783 A. polynesiensis
Mello et al. (2008)	Conversão do sistema RGB para o sistema Limiarização adaptativa (no canal H) Análise de componentes conectados Conversão do sistema RGB para o sistema Limiarização simples no canal I (YIQ) Operações morfológicas de fechamento Conversão do sistema RGB para o sistema	6 imagens	EM = 8,66% Método 1 EM = 12,64% Método 2 EM = 7,39% Método 3
Santos et al. (2008)	Conversão do sistema RGB para o sistema Limiarização simples no canal I (YIQ) Análise de componentes conectados Operações morfológicas de fechamento Conversão do sistema RGB para o sistema	6 imagens	EM = 10,69% Método 1 EM = 7,33% Método 2 EM = 7,84% Método 3

(Continuação)

Trabalho	Técnica	Validação	Principais Resultados
Gusmão, Machado e Rodrigues (2009)	Conversão do sistema RGB para o sistema Limiarização adaptativa (no canal H) Análise de componentes conectados Operações morfológicas de fechamento Conversão do sistema RGB para o sistema Limiarização simples no canal I (YIQ)	6 imagens	EM = 8,66% Método 1 EM = 12,64% Método 2 EM = 7,39% Método 3
Portela (2009)	Conversão do sistema RGB para L*a*b* Conversão do sistema RGB para o sistema Limiarização simples com um patamar Análise de componentes conectados	6 imagens	EM = 17,14%
Elpídio et al. (2010)	Quantização para tons de cinza Limiarização simples no canal R (RGB) Abertura e fechamento morfológicos Análise de componentes conectados Segmentação global com base em reg	50 imagens 50 imagens	R = 0,9298 Base A R = 0,8687 Base B
Silva e E. (2010)	Segmentação global com base em regras Segmentação local no canal R (RGB) Filtro da moda para remoção de ruídos Análise de componentes conectados	58 imagens	EM = 16,85%
Elpídio et al. (2011)	Quantização para tons de cinza Binarização de limiar único	30 imagens	EMQ = 0,09

(Continuação)

Trabalho	Técnica	Validação	Principais Resultados
Tiba e Araujo (2011)	Limiarização simples com um patamar	18 palhetas	EM = 13,94% SARCO MS1 EM = 9,72% SARCO MS2
Silva, Rodrigues e Araujo (2012)	Segmentação global com base em duas regras Segmentação local no canal R (RGB) Filtro da moda para remoção de ruídos Operação de abertura para remoção dos ruídos Análise de componentes conectados	100 imagens	EM = 16,26%
Brasil et al. (2015)	Segmentação de foreground e background Dilatação e erosão morfológicas	50 imagens	Comparação de médias
Feitosa et al. (2015)	Normalização do histograma no canal R (RGB) Limiarização de Otsu no canal R (RGB) Operação morfológica de abertura Análise de componentes conectados Conversão do sistema RGB para L*a*b*	5 imagens 5 imagens	EM = 24,03% Método 1 EM = 17,02% Método 2
Morais et al. (2015)	Limiarização simples com um patamar Análise de componentes conectados	4 imagens	EM = 3,89%
Alves (2016)	Limiarização simples no canal R (RGB) Limiarização simples no canal B (RGB) Análise de componentes conectados Limiarização simples com um patam	48 palhetas 30 algodões	EM = 28,39% EM = 15,84%

(Continuação)

Trabalho	Técnica	Validação	Principais Resultados
GABURRO et al.	Limiarização simples com um patamar Análise de componentes conectados	48 imagens	R = 0,9823 R = 0,9823
Rizzi et al. (2016)	Conversão do sistema RPG para HSV Limiarização simples no canal H Análise de componentes conectados	5 imagens	EM = 27,01%
Costa (2017)	Equalização para tons de cinza Limiarização adaptativa baseada em descontinuidade Segmentação dos ovos e bounding-boxes Extração de características para treinamento	206 imagens	Acurácia = 98,94%
Yussof et al. (2018)	Filtros de gabor para realce Limiarização adaptativa de Otsu Abertura e dilatação morfológicas Análise de componentes conectados	5 imagens 5 imagens	EM = 1,49% macro EM = 0% micro
Garcia et al. (2019)	Segmentação de regiões externas à palheta Conversão do sistema RGB para L*a*b*	30 imagens 27 imagens	R = 0,86 R = 0,9703
Rocha, Bizerra e Magalhães (2019)	Realce por filtro passa-alta (High-boost)	1 palheta	Taxa de recall = 60%

(Continuação)

Trabalho	Técnica	Validação	Principais Resultados
Junior et al. (2019)	Segmentação de regiões externas à palheta Conversão do sistema RGB para L*a*b*	15 imagens 8 palhetas	EMA = 5,14% EMA = 16,2%

Fonte: Adaptado de (BRUN et al., 2020)

Pela Tabela acima, é possível notar que a comunidade científica vem desenvolvendo trabalhos abordando o problema da contagem de ovos de *Ae. aegypti* com diferentes técnicas. Os primeiros trabalhos foram publicados no ano de 2008 e os mais recentes são do ano de 2019, ou seja, ainda há uma atividade contínua para buscar melhores técnicas e melhores soluções que possibilite a automação desta tarefa.

3.1 CONCLUSÃO DO CAPÍTULO

Os trabalhos relacionados apresentados mostram os desafios da contagem de ovos de forma automática e a problemática de criar uma base de dados representativa. A maioria deles trabalharam com uma base de dados com poucos exemplos e, conseqüentemente, são avaliados em conjuntos pequenos de dados. Fica clara a necessidade da disponibilização de uma base de dados que possibilite que os trabalhos na área avancem de maneira mais célere.

Portanto, dado o contexto dos trabalhos relacionados e suas dificuldades, esta pesquisa tem como ponto positivo a disponibilização de uma base de dados robusta de imagens e um dispositivo de *hardware* que facilita o processo de aquisição e incrementação dessas bases de dados. Outro ponto positivo deste trabalho é a utilização de métodos inteligentes com Redes Neurais Profundas para fazer a automatização do processo de contagem de ovos de *Ae. aegypti*. Com isso, espera-se maior agilidade no processo de contagem, facilitando a coleta de informação sobre a densidade da população do mosquito e auxiliando no processo de tomada de decisão dos técnicos na aplicação de estudos e políticas de controle do mosquito. O próximo capítulo abordará os passos desta pesquisa e os métodos utilizados para alcançar os resultados.

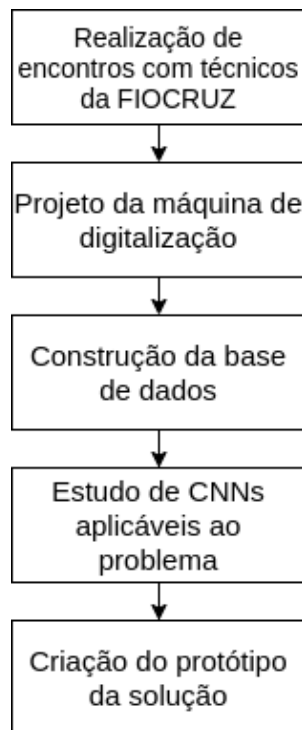
4 MATERIAIS E MÉTODOS

Este capítulo tem como objetivo contextualizar sobre os materiais utilizados para o desenvolvimento da pesquisa, e os principais métodos aplicados.

4.1 FLUXO DE DESENVOLVIMENTO DA PESQUISA

A realização desta pesquisa seguiu 5 etapas que são mostrada na Figura 15 a seguir:

Figura 15 – fluxo



Fonte: Elaborada pelo autor.

Inicialmente foram realizadas reuniões com técnicos do Instituto Aggeu Magalhães (FIOCRUZ-PE) com o objetivo de entender o processo de contagem de ovos do mosquito *Ae. aegypti*, sua relevância e gargalos desta atividade. Na reunião também foi pontuado sobre abordagens computacionais que poderiam ser utilizadas para automatizar o processo.

O segundo passo, depois de adquirida experiência do processo acima mencionado, foi projetar um dispositivo para digitalização de palhetas de ovitrampas. O dispositivo foi criado para capturar imagens das palhetas e armazená-las em um computador.

No terceiro passo foi dado início a construção da base de dados, feita em conjunto com os técnicos da FIOCRUZ-PE, eles deram um treinamento de como identificar as características

do ovo do *Ae. aegypti*, Além de validarem se as anotações feitas nas imagens estavam corretas.

No quarto passo, foi realizada a pesquisa de modelos de reconhecimento de padrões que poderiam ser aplicados para o reconhecimento de ovos em imagens.

Por fim, no quinto passo foi criado um protótipo com o dispositivo de digitalização construído neste projeto e uma GPU para fazer a contagem automática de ovos.

A seguir são serão detalhados os os materiais e métodos utilizados no projeto.

4.2 DISPOSITIVO DE *HARDWARE* DESENVOLVIDO PARA CAPTURA DE DADOS

Em trabalhos que abordam a contagem de ovos do mosquito *Ae. aegypti* é comum a utilização de câmeras profissionais e até auxílio de microscópio, o que faz o custo total dos projetos serem altos e, muitas vezes, até de uso complexo. Através do estudo da literatura foi constatado o desafio de construir um dispositivo para digitalização de palhetas.

Neste trabalho foi projetado um dispositivo de baixo custo para a digitalização de palhetas utilizando estrutura de uma impressora 3D(sem a cabeça de impressão) que recebe comandos de um Arduíno para movimentar uma câmera de microscopia de 5 pixeis. Cada imagem capturada tem 680 x 480 de dimensão. A câmera e estrutura de impressora 3D são de baixo custo o que viabiliza a produção do protótipo para uso em laboratórios. As Figuras 16 - (a), 16 - (b) e 16 - (c) abaixo mostram o equipamento em diferente ângulos.

Figura 16 – Dispositivo de digitalização de palhetas de ovitrampas desenvolvido neste trabalho sendo mostrado de três ângulos.



(a) Frente do dispositivo

(b) Dispositivo visto de cima

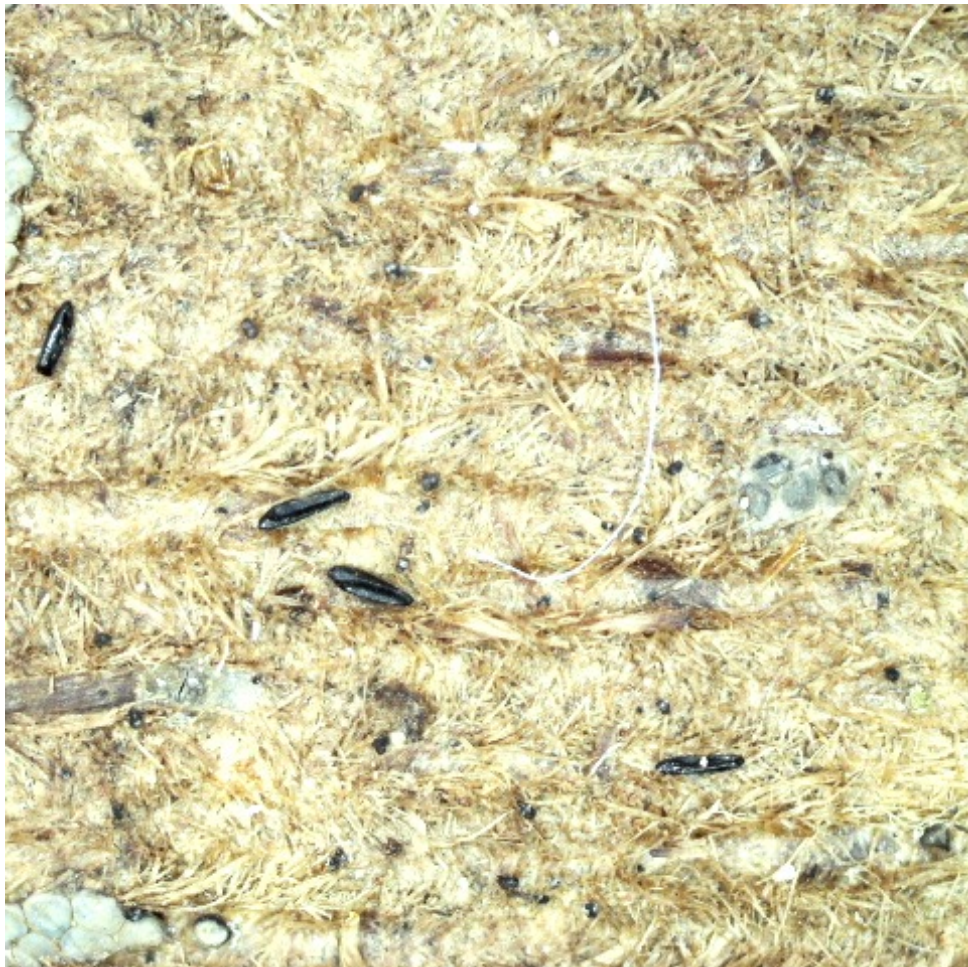
(c) Lateral do dispositivo

Fonte: Elaborada pelo autor (2021).

No centro do dispositivo fica a bandeja de palhetas. Na lateral fica o Arduíno que controla

os movimentos dele. Em cima fica a câmera que se movimenta para captar imagens de todas as partes da palheta. Ao fim do processo são geradas 204 imagens que possibilitam toda a cobertura da palheta. A Figura 17, abaixo, mostra uma imagem gerada pelo dispositivo. Mesmo os ovos de *Ae. aegypti* sendo bem pequenos medindo cerca de 0,4 milímetros o que os tornam difíceis de serem observados sem auxílio de uma lente de aumento, com a câmera do dispositivo é possível observar que eles ficam bem visíveis.

Figura 17 – Imagem coletada com o dispositivo digitalizador de palhetas. Na imagem é possível ver nitidamente os ovos.

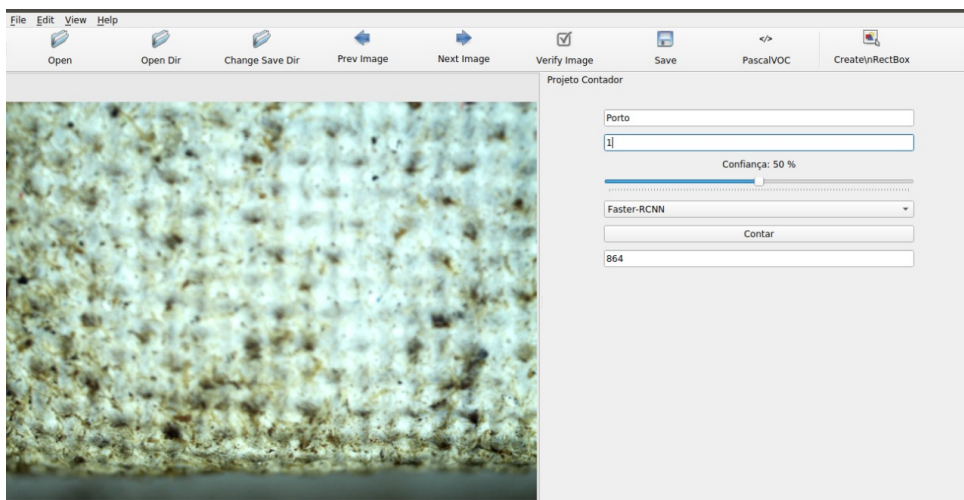


Fonte: Elaborada pelo autor (2021).

Para o dispositivo ser de fácil manuseio, foi desenvolvido um *software* para que o usuário não precise se preocupar em configurá-lo, ele só precisa colocar a palheta no centro do dispositivo de digitalização de palhetas, e por meio da interface do *software* clicar em um botão para que o dispositivo faça a digitalização completa. O *software* é criado a partir de uma ferramenta de código aberto chamada *labelimg* que foi utilizada neste projeto para o processo de anotação das imagens na base de dados. O reaproveitamento da plataforma fez sentido por ela ter

uma parte visual interativa com as imagens. Para se adaptar ao problema de contagem de ovos foram desenvolvidos módulos específicos para que o processo de anotação não fosse feito mais por um humano, mas sim por um rede neural e a intervenção humana seja só para uma checagem e correção de possíveis erros cometidos pela rede. A anotação automática é feita a partir da inferência do modelo de rede neural, com a resposta da rede a uma imagem, criamos um *script* que gera um arquivo de rotulação para a imagem, esse arquivo pode ser editado via interface quando o usuário cria ou deleta uma rotulação feita pela rede. A Figura 18 mostra a interface do sistema.

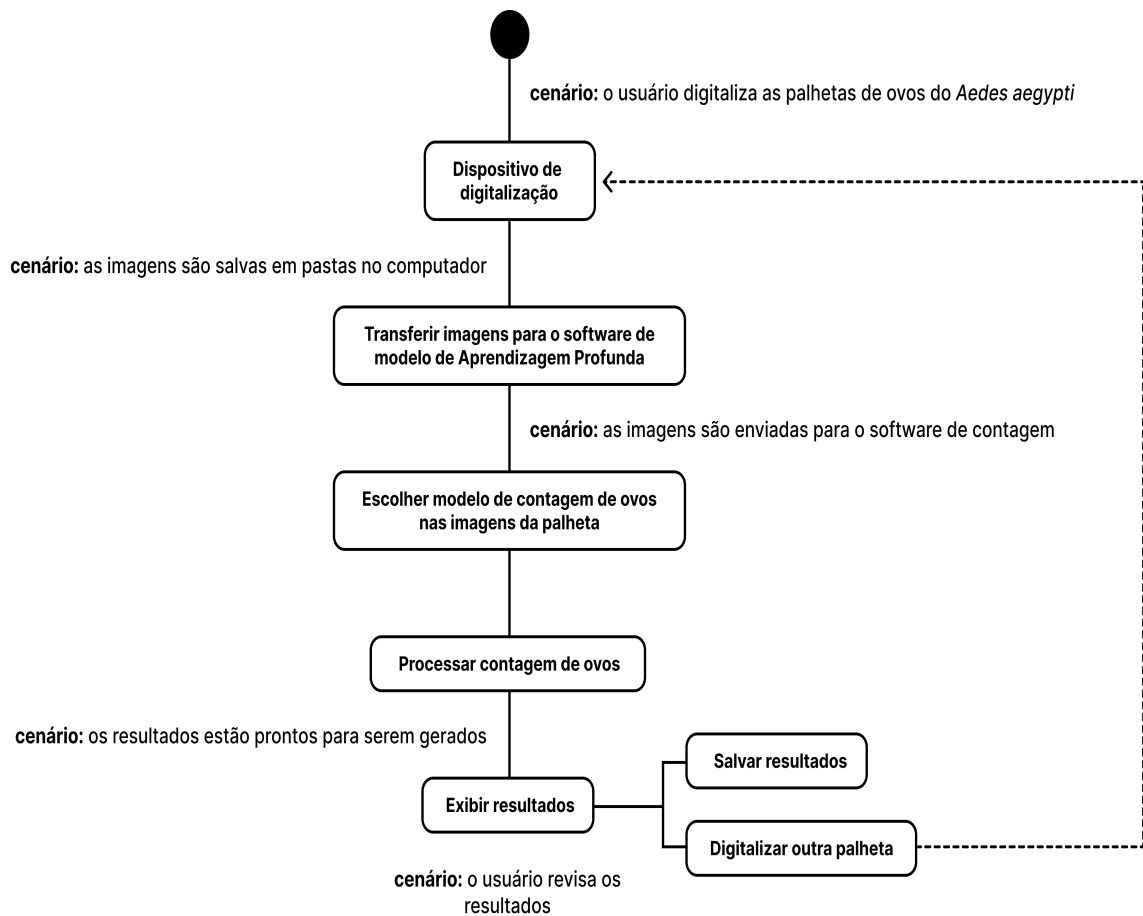
Figura 18 – Interface do *Software*. Nesta interface o usuário pode acionar a máquina para digitalizar a palheta e fazer a contagem de ovos automática com o modelo de CNN selecionado.



Fonte: Elaborada pelo autor (2021).

Na interface do *software*, o usuário digita um nome da localidade da palheta e a numeração dela, depois ele escolhe qual modelo de detecção fará a contagem da palheta e qual a confiança que ele deseja, ou seja, o usuário definirá qual a probabilidade do modelo dizer que determinada detecção é um ovo. Depois que a rede faz a contagem, o usuário pode rever a contagem utilizando o sistema e revisar suas rotulações quando for necessário, assim, ao final da revisão o sistema cria um relatório com os dados de contagem da palheta e enviará as imagens com suas anotações para incrementação da base de dados. A Figura 19 mostra o fluxo de utilização do dispositivo usando um diagrama de modelagem de interação MoLIC.

Figura 19 – Fluxo para utilização do dispositivo para digitalização, contagem e revisão de resultados.



Fonte: Elaborada pelo autor (2021).

O primeiro passo para a utilização do dispositivo é o fornecimento da palheta para ele, em seguida via *software* o técnico envia o comando para iniciar a digitalização, ao término da digitalização o modelo de CNN acoplado no protótipo recebe todas as imagens referentes a palheta e inicia a contagem. O *software* que controla o dispositivo que faz a digitalização da palheta e a contagem, pode ser instalado tanto em um computador como em uma GPU, a diferença é que em uma GPU ele será mais rápido para fazer a contagem.

A principal vantagem do dispositivo é a de automação do processo, sem uma ferramenta como essa, os técnicos utilizam uma lupa e um contador manual para fazer o processo de contagem, tornando-se um procedimento estressante e cansativo com o passar do tempo, aqui eles têm a oportunidade de colocar o dispositivo para fazer a digitalização e contagem da palheta enquanto ele se concentra em outras categorias de atividades que possam agregar mais valor ao seu trabalho. Outro benefício é que os técnicos conseguem armazenar as informações de modo que eles tenham fácil acesso quando precisar obtê-las.

A desvantagem do dispositivo é que por ser apenas um protótipo, ele tem capacidade para

apenas uma palheta, fazendo assim, o técnico ter que colocar uma palheta a cada 9 minutos, pois para a digitalização de uma palheta o dispositivo captura 204 imagens, para cada imagem há um delay de 3 segundos, esse tempo é importante para a estabilização do foco da câmera. Portanto, para a digitalização de uma palheta é necessário um tempo de 9 minutos. O ideal seria que o dispositivo tivesse mais um compartimento para acoplar mais palhetas para que os técnicos tenham um maior aproveitamento do tempo.

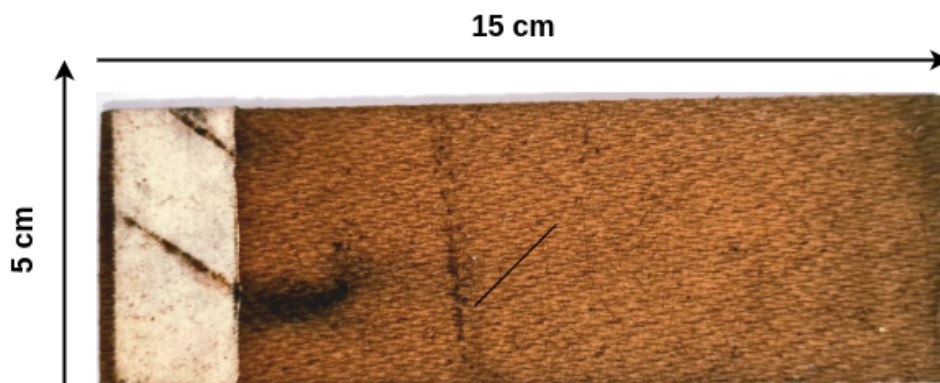
4.3 BASE DE DADOS DE IMAGENS

Um dos grandes desafios para a criação de um método inteligente para a contagem automática de ovos de *Ae. aegypti*, é a construção de uma base dados representativa que possibilite que os modelos de inteligência artificial reconheçam o padrão dos ovos para conseguirem detectar corretamente nas imagens. Os trabalhos na literatura utilizam base de dados pequenas, pois é complexo conseguir imagens de qualidade devido à dificuldade de se trabalhar com microscopia. Nesta Subseção serão relatados os passos para a construção de base de dados com imagens utilizadas neste projeto.

4.3.1 Aquisição das palhetas

Para a realização do projeto, foi feita uma parceria com a FIOCRUZ-PE. A instituição disponibilizou palhetas retiradas de diferentes localidades do Estado de Pernambuco. Foram fornecidas 100 palhetas. A Figura 20 mostra um exemplo de palheta coletada de uma ovitrampa em Pernambuco.

Figura 20 – Palheta e suas dimensões.



Fonte: Elaborada pelo autor (2021).

Estas dimensões de palhetas mostradas na Figura 20 se ajustam perfeitamente ao dispositivo criado neste trabalho para fazer a digitalização.

Além de fornecer as palhetas, foram feitas reuniões com técnicos sobre a problemática da contagem e os desafios para a resolução do problema. Também foi dado um treinamento para ajudar na identificação dos ovos em imagens. O treinamento foi importante para o processo de anotação das imagens utilizadas pelos modelos de reconhecimento de padrões. O processo de anotação é explicado na próxima seção.

4.3.2 Anotação das imagens

O processo de anotação de dados é uma etapa importante quando se trata de utilizar sistemas de IA, algumas vezes, já se tem bases de dados prontas para determinados problemas e os pesquisadores gastam seu tempo construindo o modelo de IA, entretanto, quando não há uma base de dados de qualidade, é necessário construir uma própria, esse processo leva bastante tempo a depender do número de pessoas envolvidas. Neste projeto toda a anotação foi feita por apenas um pesquisador. Entenda como anotação de dados o processo de marcar na imagem o ovo, dada essa marcação é passada para o modelo IA a imagem e arquivo de marcação para ele entender onde se encontra o que ele deve aprender o padrão.

Dado o treinamento pelos técnicos, foi feito o processo de anotação das imagens que se dá marcando cada ovo na imagem. Essa parte do processo foi feita totalmente manual até se construir um modelo de base. A partir do modelo base o processo foi feito de forma semiautomática. Ou seja, o modelo cria as anotações automaticamente, mas foi corrigido manualmente em caso de erros do modelo. As anotações foram feitas na ferramenta de *labeling*¹. A Figura 21 abaixo mostra como é o processo de anotação de imagens, nela há 3 ovos sendo marcados. O resultado deste processo é um arquivo contendo a localização de cada ovo e suas dimensões.

¹ <https://github.com/tzutalin/labelImg>

Figura 21 – Exemplo de anotação de imagem. Ao centro é possível observar o processo de criação de caixas nos ovos. Cada caixa marca as coordenadas x e y, tais informações são importantes para os modelos de detecção buscarem especificamente nas imagens o que eles tem que aprender e onde.



Fonte: Elaborada pelo autor (2021).

Na anotação das imagens foram encontrados desafios que precisaram ser considerados para construir uma base consistente. Um dos pontos notados foi que apesar do dispositivo capturar 204 imagens de cada palheta, grande parte dessas imagens não apresentavam nenhum ovo e precisaram ser descartadas, há uma justificativa para a falta de ovos em toda a palheta, uma delas é que ela não é posta completamente na água na ovitrampa, logo apenas parte de palheta vai conter ovos, entretanto foi verificado em análises das imagens que os ovos podem estar em qualquer lugar da palheta. Portanto, foi decidido fazer a digitalização de toda a palheta para não correr o risco de deixar imagens que continham ovos de fora. Como o objetivo é fazer o modelo reconhecer os ovos, foram utilizadas apenas imagens que apresentava pelo menos um ovo.

4.3.3 Pré-processamento das imagens

A etapa de pré-processamento de imagens é importante para buscar melhorar a qualidade das imagens que serão dadas como entradas para os modelos de CNN.

Neste projeto algumas imagens se apresentaram desfocadas, pois, algumas vezes, palhetas tinham formato irregular devido ao tempo que elas ficaram nas ovitrampas. Com isso, a câmera do dispositivo de digitalização de palhetas não conseguiu fazer uma boa captura de imagens em determinados pontos de algumas palhetas.

No caso de imagens desfocadas, ao tomar a decisão de anotá-las, tem-se um problema com o aumento de falsos positivos em imagens que apresentavam uma boa qualidade de foco, assim, se tem um *trade-off*, se anotar imagens desfocadas há um aumento no número de falsos positivos, se deixar de anotar imagens desfocadas o modelo terá mais dificuldade de marcar um ovo em imagens de baixa qualidade, logo, como o número de imagens desfocadas é menor que o número de imagens de boa qualidade, foi decidido focar a anotação em imagens que apresentavam um bom padrão para, assim, diminuir o número de falsos positivos. Este processo de escolha das imagens em relação ao foco foi realizado de forma manual, cada imagem foi analisada antes de fazer o processo de anotação.

Além do problema do desfoque das imagens, também foram encontrados problemas de iluminação. Nestes casos, foram utilizados métodos tradicionais e inteligentes de visão computacional para contornar este problema. Foram utilizados correção de iluminação com contraste e normalização de histograma para ajudar na visualização dos ovos nas imagens. As figuras 22 e 23, abaixo, mostram o processo de normalização de histograma em imagem.

Figura 22 – Imagem original antes da normalização de histograma. Como é possível observar, a imagem é bastante escura dificultando a visualização do ovo na parte inferior.

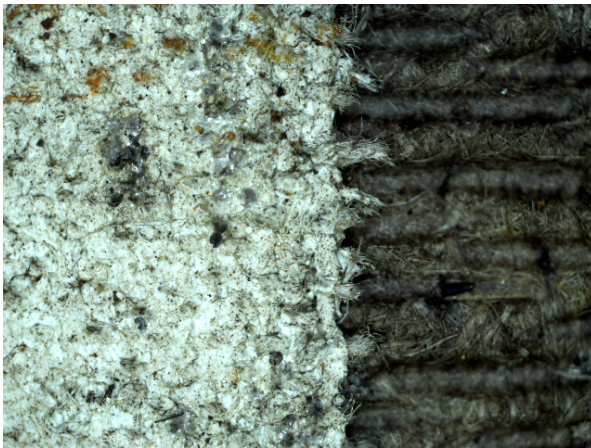
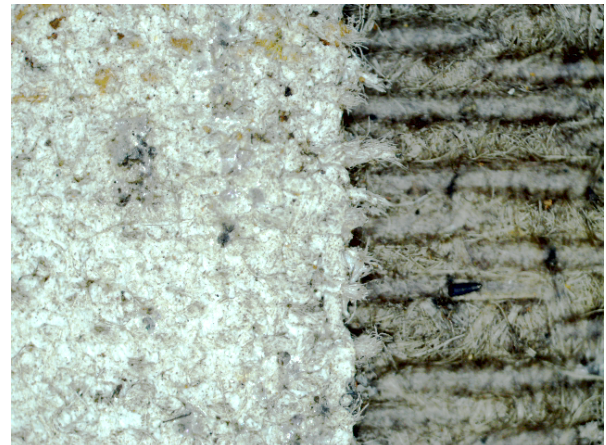


Figura 23 – Imagem após a normalização de histograma. Já nesta imagem é possível observar que o ovo aparece na parte inferior. Tal melhoria, já faz os modelos reconhecerem o padrão do ovo.



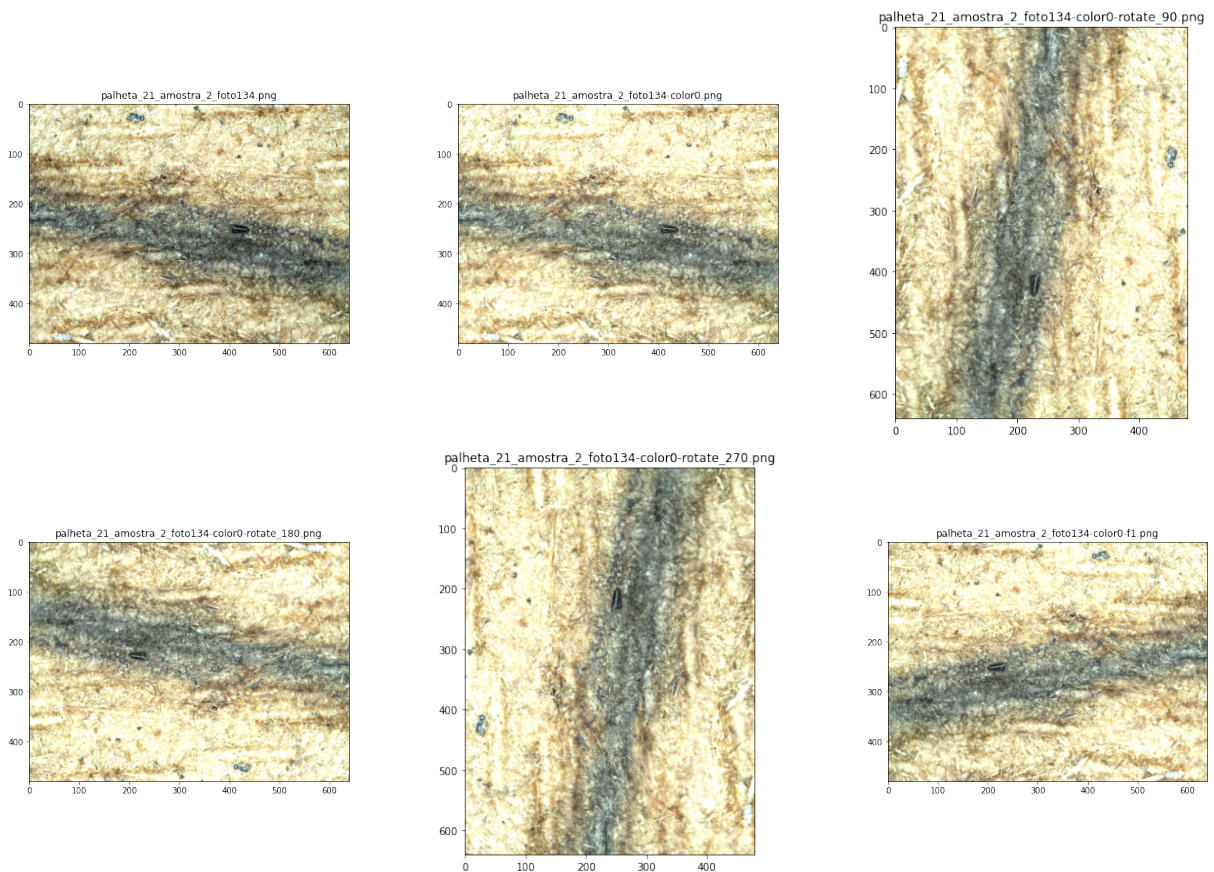
Fonte: Elaborada pelo autor (2021).

O problema de iluminação acontece em algumas palhetas de cores mais escuras, ou também devido à quantidade de sujeira que pode fazer o ovo ficar em pontos que diminuem sua visibilidade. Portanto, é interessante tratar esse problema para um melhor resultado na detecção. Apesar de fazer isso em algumas imagens, foi decidido anotar esses padrões escuros também para uma melhor generalização dos modelos.

Outra técnica aplicada na construção da base foi a de aumento de dados, do inglês,

Data Augmentation. Esta técnica permite a criação de dados a partir de dados existentes, ou seja, uma imagem pode ser representada de diferentes formas, os benefícios desta técnica é o aumento no número de imagens na base com menos custo de trabalho na construção da mesma, além de focar em imagens de melhor qualidade para o modelo. A Figura 24 a seguir mostra a transformação de uma imagem no processo de aumento de dados. A primeira imagem é a original e as seguintes são modificações da original, as principais modificações são rotação e iluminação.

Figura 24 – Processo de aumento de imagens a partir de uma imagem original, são feitas novas imagens aplicando filtros e rotações. Nota-se que o ovo presente na imagem vai se deslocando a medida que as aplicações de aumento de dados são aplicadas.



Fonte: Elaborada pelo autor (2021).

Foram selecionadas 529 imagens aleatórias provenientes de diferentes palhetas para aplicar o processo de aumento de dados. A partir destas imagens foi construído uma base de dados com 3703 imagens. A seção seguinte mostrará como essas imagens foram organizadas para realizar o treinamento e testes dos modelos utilizados.

4.3.4 Divisão da base de dados

Uma etapa necessária antes da aplicação dos algoritmos de reconhecimento de padrão é a divisão da base de dados. Um método comum na literatura é o *Hold-out validation*. Esta técnica sugere dividir os dados em três conjuntos, são eles: treino, validação e teste. O primeiro consiste nos dados utilizados para o treino do modelo, o segundo é utilizado para ajuste dos hiperparâmetros do modelo, e o terceiro é utilizado para mensurar o desempenho do modelo. Neste trabalho dividimos a base de dados utilizando 80% das imagens para treinamento, 10% para validação e 10% para teste. A Tabela 2 mostra a divisão da base de dados e o número de anotações de ovos feitas em cada divisão.

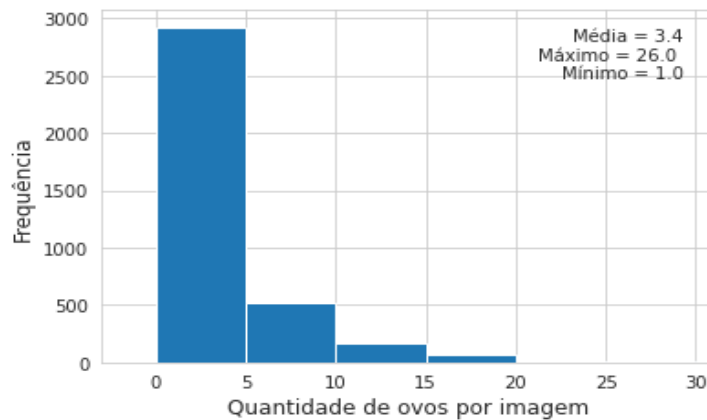
Tabela 2 – Divisão da base de dados entre dados de treinamento, validação e teste. Cada linha da tabela apresenta a quantidade de imagens no conjunto de dados e a quantidade total de ovos que tem nas imagens.

Divisão da base de dados		
Divisão do dataset	Número de imagens	Número de ovos
Treino	2912	9737
Validação	399	1862
Teste	392	1106

Fonte: Elaborada pelo autor (2021).

Como já foi mencionado, nem todas as palhetas e nem todas as imagens apresentam ovos. Como neste projeto só é utilizado imagens com ovos para ensinar os modelos a detectá-los, foi realizada uma análise da quantidade de ovos que aparece por imagens na base de dados. A Figura 25 mostra o resultado dessa análise.

Figura 25 – Histograma do número de ovos em imagens. As imagens são da base de dados construída neste trabalho. Grande parte das imagens apresenta de 1 a 10 ovos.



Fonte: Elaborada pelo autor (2021)

Como é possível (constatar) na Figura 25, a maior parte das imagens contém entre 1 e 10 ovos, poucas imagens apresentam número de ovos acima de 15. A quantidade máxima de ovos em uma imagem foi 26 e a mínima foi de 1 para atender a regra de ter pelo menos um ovo por imagem.

Um dos objetivos deste trabalho é publicar essa base de dados para colaborar com outros trabalhos que buscam resolver a mesma problemática, dado que, construir uma base de dados foi um problema constatado nos trabalhos relacionados devido à dificuldade de captura de imagens de ovos de *Ae. aegypti*. A base de dados utilizada neste trabalho pode ser baixada em: <https://drive.google.com/file>.

Na próxima seção serão apresentados os modelos de CNNs utilizados no projeto que foram treinados com esta base de dados.

4.4 MODELOS DE CNNs UTILIZADOS NESTE TRABALHO

Tendo a base de dados finalizada, o passo seguinte foi a escolha de modelos de CNN para adaptar ao problema de contagem de ovos de *Ae. aegypti* em imagens. Para a escolha dos modelos foram utilizados dois critérios: desempenho em competições de visão computacional e velocidade de processamento.

O desempenho desses modelos são testadas em problemas de detecção de objetos que utilizam a Precisão Média (do inglês *Average precision* — AP) como métrica de desempenho, além de analisar também a velocidade de processamento dessas redes, ou seja, o tempo

que elas levam para dar uma resposta de detecção uma imagem. As principais competições de detecção de objetos como *VOC PASCAL Challenge*, *COCO*, *ImageNet Object Detection Challenge* e *Google Open Images Challenge* utilizam essa métrica para avaliar os modelos em seus conjuntos de dados de imagens (PADILLA; NETTO; SILVA, 2020).

A detecção de objetos em imagens trata-se de identificar e localizar alguns objetos de interesse tais como: carros, ônibus, frutas, pessoas, etc. Neste caso, a AP mensura o quão bem a rede está detectando/localizando o objeto de interesse na imagem a partir da caixa criada pelo modelo (PADILLA; NETTO; SILVA, 2020). A Figura 26 mostra um exemplo de detecção de objetos em imagens.

Figura 26 – Exemplo de detecção de uma CNN. No exemplo a rede aplica a predição na imagem e cria duas caixas, uma, no homem e outra no cachorro. Além das caixas a rede também mostra a probabilidades para cada caixa.



Fonte: (PADILLA; NETTO; SILVA, 2020)

Na Figura 26 o modelo detecta uma pessoa e um cachorro, e atribui uma probabilidade de acerto para cada. Neste exemplo, a pessoa representa uma classe de objeto e o cachorro outra, logo é possível calcular um AP médio (do inglês *Mean Average precision* - mAP) de todas as classes que aparece na imagem. Os melhores modelos na literatura conseguem bons resultados de mAP, dessa forma foi utilizada essa métrica como norte para escolher os modelos.

Dado este contexto, foi decidido utilizar os modelos: EfficientDet, YOLOv4 e RFCN explicados com mais detalhes no referencial teórico.

4.5 TREINAMENTO DOS MODELOS CNNs UTILIZADAS

Tendo a base de dados finalizada, o passo seguinte é treinar os modelos de CNNs para fazer o reconhecimento de ovos de *Ae. aegypti* em imagens. Nesta etapa é passado para o modelo os dados que ele utilizará para aprender os padrões (base de treinamento) e os dados que ele utilizará para ajustar seus parâmetros (base de validação). Para cada modelo é necessário configurar seus parâmetros para que eles se adéquem aos dados que estão sendo fornecidos, esse processo é chamado *tuning*. Os modelos aprendem os padrões fornecidos durante o treinamento. Em cada imagem fornecida ele vai aprendendo os padrões de interesse até que seu erro ao detectar um padrão seja mínimo.

A Figura 27 abaixo mostra um modelo que ainda não aprendeu o padrão do ovo corretamente fazendo uma inferência em uma imagem de exemplo. Como no início do treinamento a rede ainda não aprendeu nenhum padrão, ela marca quase todas as partes da imagem como sendo ovos.

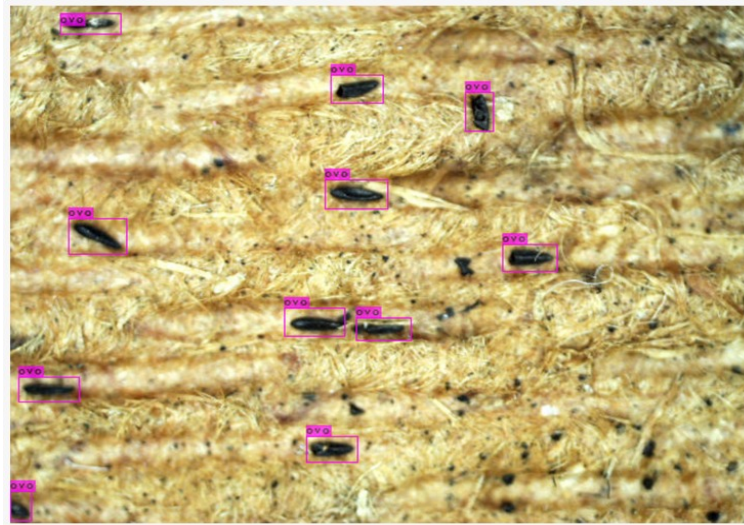
Figura 27 – Início de treinamento do modelo YOLOv4 que começa criando a quantidade de caixas possível, pois ainda não aprendeu a reconhecer os ovos.



Fonte: Elaborada pelo autor (2021).

Já a Figura 28 abaixo apresenta a mesma imagem ao modelo no final do seu treinamento e ele consegue marcar com perfeição cada ovo que presente na imagem.

Figura 28 – Fim do treinamento do modelo. Na Figura é possível notar que agora o modelo marca apenas os ovos que aparecem na imagem, ou seja, o modelo aprendeu a reconhecer corretamente o padrão do ovo.



Fonte: Elaborada pelo autor (2021).

O processo de treinamento de um modelo de CNN é demorado, cada rede testada neste trabalho leva entre 5 e 10 horas para finalizar o seu treinamento. Por se tratar de um processo que demanda recursos computacionais de GPU, o treinamento destes modelos foi realizado na plataforma *Google Colab*², pois o *Google Colab* fornece uma GPU Tesla K80 suficiente para treinar modelos de redes neurais, além de permitir criar *Scripts* em linguagem de programação *Python* e fornecer integração com a plataforma *Google Driver* para salvar todos os arquivos necessários do projeto como: base de dados, arquivos de programas e relatórios. Alguns dos *scripts* utilizados para treinar as redes podem ser acessados em: <https://cutt.ly/SR9T4CH>. A seguir serão apresentadas as métricas para avaliar se os modelos estão ou não conseguindo aprender os padrões fornecidos.

4.6 MÉTRICAS UTILIZADAS PARA AVALIAR OS RESULTADOS DOS MODELOS DE CNN ESCOLHIDOS

As métricas são importantes para mensurar a qualidade dos resultados obtidos pelos modelos verificando o acerto e o erro dos mesmos, existem diferentes métricas para cada problema (SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006). E uma das formas visualizar as métricas é a partir da matriz de confusão. Uma matriz de confusão compara as previsões de um modelo

² <https://colab.research.google.com/>

com os valores verdadeiros. A diagonal da matriz representa classes que foram corretamente previstas sendo Verdadeiro Positivo (VP) ou Verdadeiro Negativo (VN), enquanto elementos fora da diagonal representam indivíduos que são classificados de forma errada sendo Falso Positivo (FP) ou Falso Negativo (FN). A Tabela 3 mostra um exemplo de matriz de confusão.

Tabela 3 – Matriz de Confusão.

Classe Verdadeira	Classe Predita	
	Positiva	Negativa
Positiva	VP	FN
Negativa	FP	VN

Fonte: Elaborada pelo autor (2021).

Em problemas de classificação binária as métricas Acurácia, Precisão e Sensibilidade são amplamente utilizadas para mensurar a qualidade de predição dos modelos.

A Acurácia é o número de previsões corretas feitas pelo modelo sobre todas as categorias de previsões realizadas.

Para calcular a acurácia de um algoritmo é utilizada a seguinte fórmula:

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \quad (4.1)$$

A Precisão é a razão de previsões corretas (VP) para o total de observações positivas previstas da classe.

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (4.2)$$

A Sensibilidade é a razão do número de exemplos classificados como pertencentes a classe verdadeira que realmente são da classe verdadeira.

$$\text{Sensibilidade} = \frac{VP}{VP + FN} \quad (4.3)$$

Também foram utilizadas as métricas RMSE, do inglês *Root Mean Square Error*, e MAE, do inglês *mean absolute error*, por ser as métricas mais populares em trabalhos de contagem de ovos de mosquito *Ae. aegypti* para avaliar a taxa de erro. Ambas, permitem mensurar o erro entre a contagem manual e a contagem automática dos modelos.

RMSE é a métrica que calcula a raiz quadrática média dos erros entre valores reais e as previsões do modelo (BONNIN, 2017).

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (y - \hat{y})^2} \quad (4.4)$$

Enquanto a métrica MAE calcula o erro absoluto (BONNIN, 2017).

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} |y_i - \hat{y}_i| \quad (4.5)$$

Onde, y_i é o valor real e \hat{y}_i é o valor predito pelo modelo.

Ambas as métricas são utilizadas em problemas de regressão, ou dizendo de outro modo, quando a variável alvo é contínua. Ou seja, se encaixa no nosso problema, pois nosso objetivo é saber o número de ovos nas palhetas.

4.6.1 Testes estatísticos utilizados

Neste trabalho foram aplicados alguns testes estatísticos para comparar os resultados entre a contagem dos modelos e a contagem humana. Os testes de hipóteses estatísticas fornecem meios que permitem rejeitar ou não rejeitar uma hipótese estatística através da evidência fornecida pela amostra (SEBASTIANI; VIALI, 2011). Existem duas categorias de testes, os paramétricos e os não-paramétricos. O teste ser paramétrico ou não-paramétrico se refere aos parâmetros populacionais como a média e ao desvio-padrão que são parâmetros que definem as populações que apresentam distribuição normal. Testes não-paramétricos não requerem conhecimento da distribuição da população (SEBASTIANI; VIALI, 2011).

O primeiro teste aplicado neste trabalho foi o de Shapiro-Wilk. Este teste avalia a hipótese nula de que os dados provêm de uma distribuição normal. Em caso de rejeição da hipótese nula, se confirma que os dados não se distribuem de acordo com uma distribuição normal (CARRASCO et al., 2020). O teste Shapiro-Wilk é importante para decidir se os demais testes serão paramétricos ou não-paramétricos.

Como resultado obtido do teste Shapiro-Wilk foi suficiente para rejeitar a hipótese nula, ou seja, os dados desta pesquisa não se disturbem como uma normal, foi optado por realizar os experimentos com testes não-paramétricos.

Portanto, neste trabalho foi utilizado o teste de Friedman que é uma alternativa não-paramétrica à ANOVA de medidas repetidas (CARRASCO et al., 2020). Este teste é usado para

determinar se existe ou não uma diferença estatisticamente significativa entre as médias de três, ou mais grupos em que os mesmos objetos aparecem em cada grupo. O Teste de Friedman usa as seguintes hipóteses nulas e alternativas:

- A hipótese nula (H_0): a média para cada população é igual.
- A hipótese alternativa: (H_a): pelo menos uma média populacional difere das demais.

Nos casos onde a hipótese nula é rejeitada, ou seja, quando há diferença significativa entre as populações, é possível fazer o teste Post-Hoc de Nemenyi, pois ele analisa quais populações especificamente apresentam diferenças (PEREIRA; AFONSO; MEDEIROS, 2015). O resultado do teste Post-Hoc de Nemenyi é uma tabela de comparações entre os pares das populações testadas. Na Seção 5.2 são apresentados os resultados de cada teste baseado nos dados.

4.7 CONCLUSÃO DO CAPÍTULO

Neste capítulo foi apresentado o fluxo de desenvolvimento da dissertação, o dispositivo para digitalização de palhetas, a forma de como foi construída a base de dados utilizada no trabalho, os modelos de CNNs utilizados e as métricas para comparação de resultados.

5 RESULTADOS DOS EXPERIMENTOS

A seguir, serão apresentados os resultados dos modelos nas bases de dados e, em seguida um comparativo entre a contagem por técnicos e os modelos utilizados.

5.1 DESEMPENHO NA BASE DE DADOS

A Tabela 4 apresenta os desempenhos dos modelos na própria base de dados, podendo assim, analisar se os mesmos estão conseguindo aprender padrões de ovos que estão sendo fornecidos para o treinamento deles. O desempenho é calculado conforme o número de ovos que existem em cada conjunto de dados e o quanto os modelos conseguem detectar corretamente.

Tabela 4 – Resultado dos modelos nos dados de treino, validação e teste.

Modelo	Base de dados	Acurácia	Precisão	Sensibilidade	RMSE	MAE
EfficienteDet	Treino	94%	82%	94%	1.45	0.65
	Validação	90%	87%	90%	1.35	0.55
	Teste	93%	84%	93%	0.89	0.55
YOLOv4	Treino	98%	92%	98%	0.52	0.30
	Validação	89%	94%	89%	1.39	0.63
	Teste	93%	93%	93%	0.35	0.29
RFCN	Treino	88%	86%	88%	1.54	0.63
	Validação	78%	88%	78%	5.51	1.22
	Teste	89%	89%	89%	0.74	0.51

Fonte: Elaborada pelo autor (2021).

Pela Tabela 4, é possível mostrar que os modelos estão conseguindo capturar bem os ovos nos conjuntos de dados ao verificar a acurácia em cada um deles. No caso da Precisão, que teve resultado mais baixo que a Sensibilidade significa que os modelos estão errando mais em Falsos Verdadeiros do que em Falso Negativos em alguns casos, ou seja, os modelos capturam mais ovos, mas parte deles, na verdade, não são. Entretanto, eles capturam quase todos os ovos existentes nos conjuntos de dados o que justifica a alta sensibilidade. Portanto, maximizar a precisão minimizará o número de falsos verdadeiros, ao passo que maximizar a Sensibilidade minimizará o número de Falsos negativos.

Pelos resultados de sensibilidade dos modelos mostrados na Tabela 4, é possível notar uma maior estabilidade no modelo YOLOv4, isso significa que este modelo consegue ser equilibrado quanto aos erros de Falsos verdadeiros e Falsos Negativos. No problema de contagem de ovos,

é importante buscar um modelo que tenha uma alta taxa de Precisão e também uma alta taxa Sensibilidade, pois tanto os erros de Falsos Verdadeiros como os de Falsos Negativos são prejudiciais para a análise de dados dos técnicos, visto que, o modelo que diz haver muitos ovos numa região sem, de fato, existir, haverá um falso alerta. Por outro lado, caso haja muitos ovos e o modelo não conseguir detectar, ocorrerá uma falta de ação para controle do mosquito.

A seguir será apresentado um comparativo entre a contagem humana e os modelos.

5.2 DESEMPENHO DOS MODELOS COMPARADOS AOS TÉCNICOS

Este experimento contou com ajuda de 5 técnicos da FIOCRUZ-PE para fazer a contagem manual de 13 palhetas. Cada técnico contou todas as palhetas com a restrição de não saber da contagem dos demais para evitar enviesamento no resultado. Além da presença dos técnicos, um pesquisador fez a contagem utilizando as imagens obtidas pelo dispositivo digitalizador de palhetas desenvolvido por esta pesquisa.

A Tabela 5 mostra a contagem de cada técnico da FIOCRUZ-PE, do pesquisador e dos modelos CNNs. Os técnicos são representados por C1, C2, C3, C4 e C5. O P1 é pesquisador deste trabalho e as 3 ultimas colunas são os modelos utilizados de CNNs.

A partir da Tabela 5, é possível notar que a contagem entre humanos apresenta bastante variação, enquanto as dos modelos, a variação é mais equilibrada. A Figura 29 apresenta esses dados em forma de gráfico de linhas para facilitar a visualização.

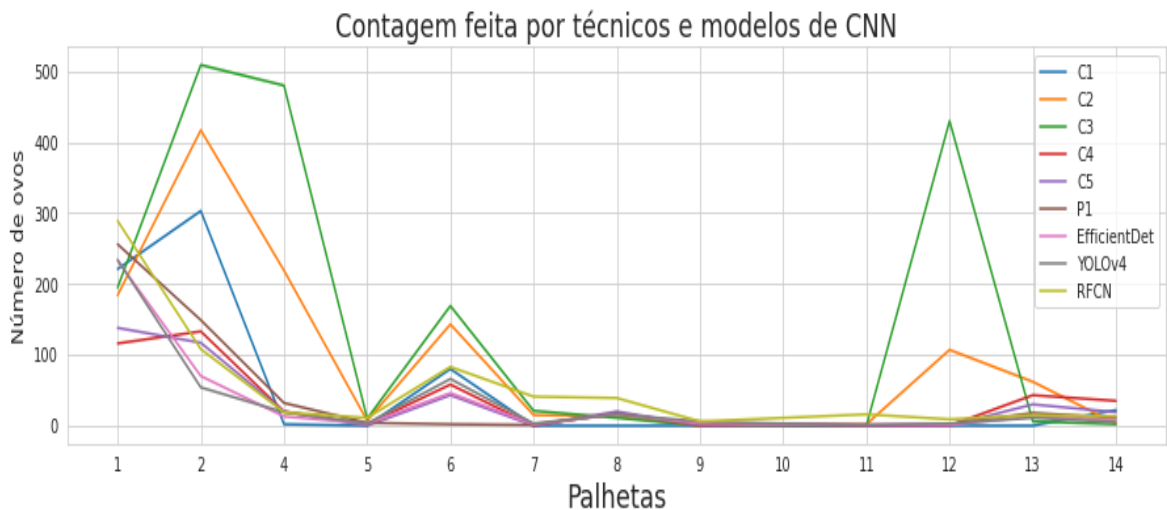
Na Figura 29 é possível notar que, apesar da variação da contagem entre os técnicos, as linhas seguem a mesma tendência. Alguns contadores como o C2 e C3 contam mais ovos que os demais. Há muitos fatores que podem contribuir para esta variação entre os técnicos, sendo o principal deles, o cansaço, que, a partir de um certo tempo contando estas palhetas, é natural que a taxa de erro aumente. Há também a dificuldade visual de mapear tudo que se observa na palheta, principalmente nas que apresentam vários ovos, pois muitos deles acabam sendo contados mais de uma vez ou alguns deixam de ser contados. Portanto, uma vantagem de utilizar aos modelos de CNNs é eliminar erros relacionados a fatores humanos como o cansaço ou distrações. Entretanto, os modelos de CNNs também apresentam erros de contagens repetidas ou algumas vezes deixam de detectar ovos presentes na imagem a depender da qualidade da imagem.

Tabela 5 – Comparação entre a contagem efetuada por pessoas e a contagem feita por modelos de CNNs.

Palheta	Contadores Humanos						Modelos		
	C1	C2	C3	C4	C5	P1	EfficientDet	YOLOv4	RFCN
1	221	184	195	116	138	256	233	234	289
2	303	417	509	133	117	149	70	54	108
3	2	219	480	20	20	32	13	19	19
4	0	7	10	2	0	4	3	3	11
5	80	143	169	58	43	74	46	66	83
6	0	15	21	0	0	1	2	3	41
7	0	13	11	18	20	17	16	16	39
8	0	0	0	0	0	2	3	6	6
9	0	1	1	3	1	1	2	3	11
10	0	2	0	0	0	1	1	1	16
11	0	107	430	0	0	2	1	3	9
12	0	62	6	43	30	18	13	11	15
13	22	3	2	35	19	11	8	5	13

Fonte: Elaborada pelo autor (2021).

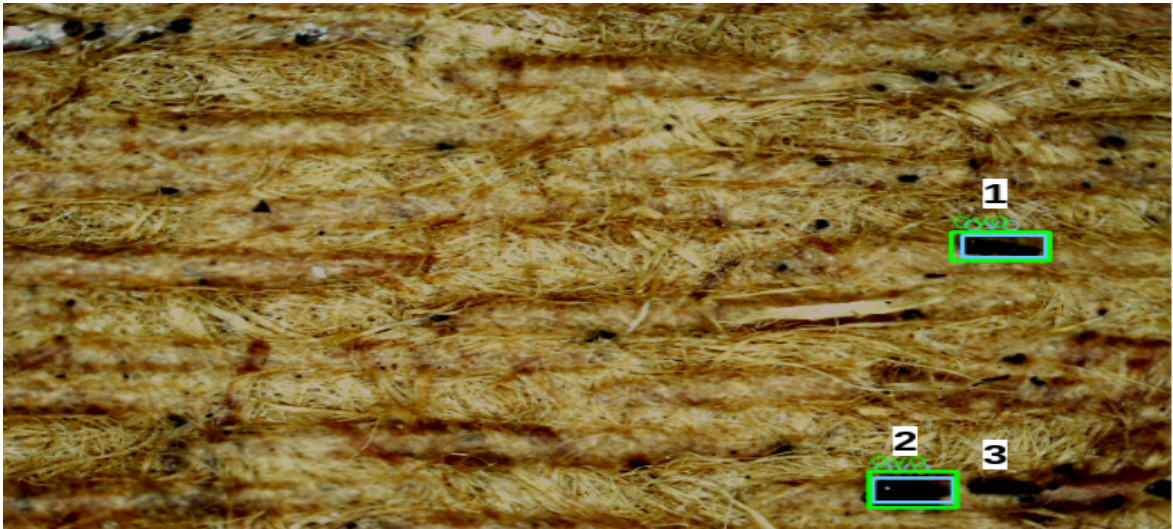
Figura 29 – Gráfico com as contagens entre humanos e modelos de CNN.



Fonte: Elaborada pelo autor (2021).

A seguir são apresentados alguns erros cometidos por modelos de CNNs e como é possível corrigi-los aplicando técnicas computacionais. A Figura 30 mostra um exemplo de detecção em que a CNN não consegue encontrar todos os ovos na imagem. Os ovos que estão dentro de um retângulo verde significa que eles foram detectados corretamente pela CNN.

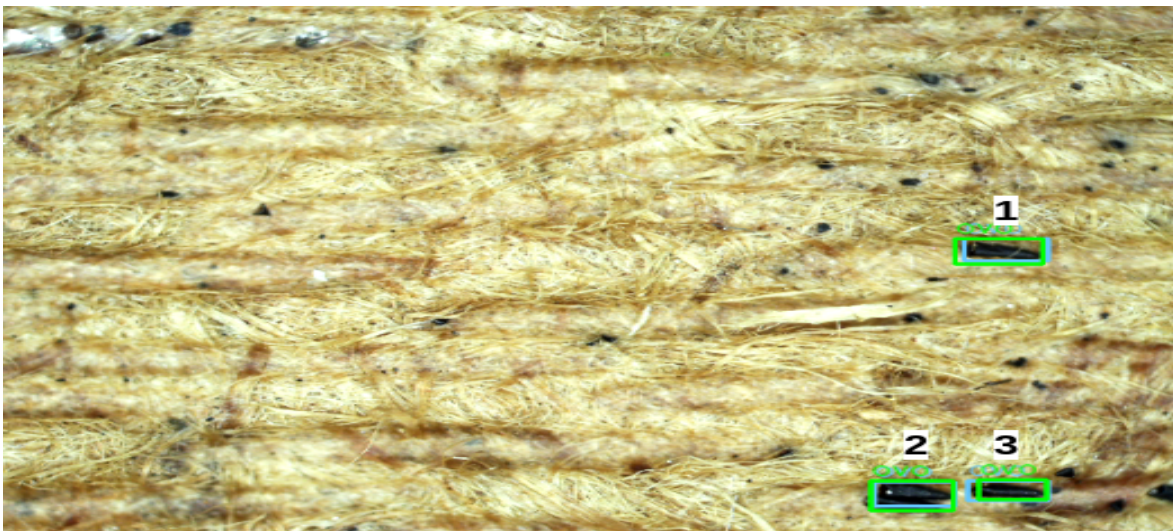
Figura 30 – Erro de detecção. Os números 1, 2 e 3 são ovos, entretanto o ovo 3 não foi detectado pela CNN.



Fonte: Elaborada pelo autor (2021)

Já a Figura 31 apresenta a mesma imagem, mas com correção de iluminação aplicado através do processo de aumento de dados. Com essa correção, a CNN consegue detectar o ovo 3, que na Figura 30 não foi detectado.

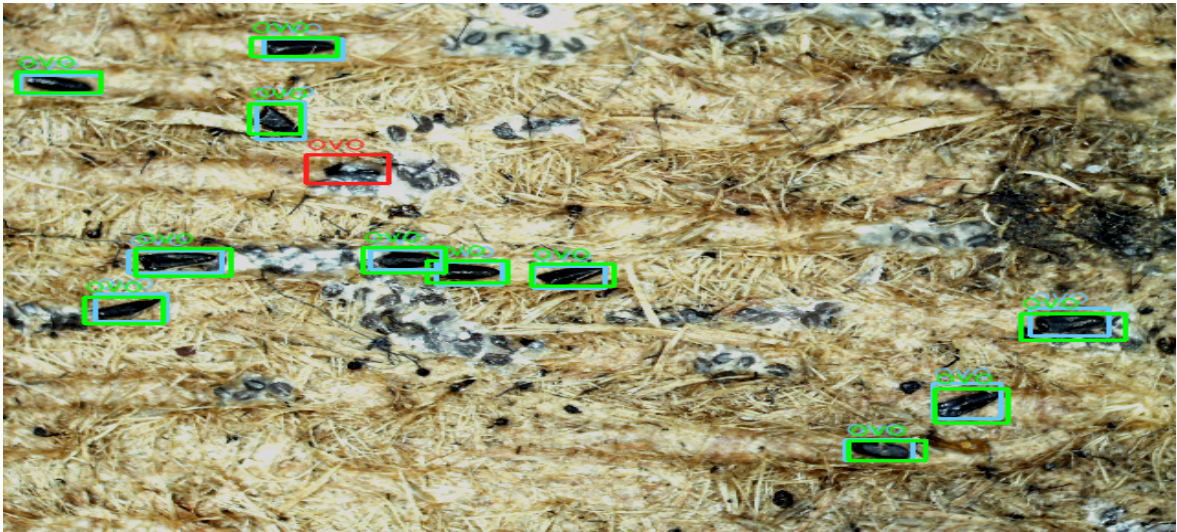
Figura 31 – Correção de detecção. Os números 1, 2 e 3 são ovos. O ovo 3 agora foi detectado corretamente pela CNN.



Fonte: Elaborada pelo autor (2021)

Outro erro que as CNNs podem cometer, é marcar algo como ovo mesmo ele não sendo um, ou seja, um falso positivo. A Figura 32 mostra um exemplo de falso positivo. Essa categoria de erro pode ser corrigida aumentando o critério da rede ao marcar um ovo e tal critério é dado por um valor de probabilidade, visto que, o resultado da rede é a marcação do ovo e a probabilidade desta marcação ser um ovo.

Figura 32 – Exemplo de falso positivo na detecção de um ovo representado pelo retângulo vermelho.



Fonte: Elaborada pelo autor (2021)

Como é mostrado nas figuras acima, as CNNs também cometem erros no reconhecimento dos ovos, entretanto, para cada categoria de erro, é possível encontrar uma solução computacionalmente viável que melhora seus resultados, exceto se a qualidade da imagem for muito baixa, de modo que não seja possível detectar nenhum padrão de ovos nela. Tais erros cometidos pelas CNNs também afetam a contagem da palheta e para verificar se as contagens entre humanos e os modelos diferem estatisticamente, foram aplicados alguns testes.

O primeiro teste aplicado foi o de Shapiro-Wilk para atestar a normalidade dos dados. A Tabela 6 apresenta os resultados deste teste. Por ela é possível constatar que nenhum contador tem distribuição normal, pois dado o valor de $\alpha = 0,05$, todos os p-valores são menores. Dessa forma, não há evidências estatísticas suficientes para afirmar que esses dados são de distribuição normal. Logo se pode prosseguir no uso de testes não-paramétricos para análise dos resultados.

A partir do resultado do teste de Shapiro-Wilk, foi aplicado o teste de Friedman e, em seguida, o teste Post-hoc de Nemeyi. Para o teste de Friedman, se formulou as seguintes hipóteses:

1. Hipótese nula (H_0): a média de contagem para cada população é igual.
2. Hipótese alternativa (H_a): pelo menos uma média populacional difere das demais.

Para o valor de $\alpha = 0,05$, a estatística de teste é 20,498 e o p-valor correspondente é 0,0086. Como esse p-valor é menor que 0,05, se rejeita a hipótese nula de que a contagem

Tabela 6 – Resultado do teste de normalidade com Shapiro-Wilk. Cada linha apresenta o contador, estatística de teste e o resultado do p-valor.

Contador	Estatística de teste	P-valor
C1	0.573	0.000
C2	0.765	0.003
C3	0.714	0.001
C4	0.760	0.002
C5	0.691	0.000
P1	0.635	0.000
EfficientDet	0.535	0.000
YOLOv4	0.539	0.000
RFCN	0.604	0.000

Fonte: Elaborada pelo autor (2021)

Tabela 7 – Comparação de p-valores entre cada par de contadores. Os resultados em vermelho são os que são menores que 0.05.

	C1	C2	C3	C4	C5	P1	Efficient-Det	YOLOv4	RFCN
C1	1.0	0.1	0.4	0.9	0.9	0.5	0.9	0.6	0.0
C2	0.1	1.0	0.9	0.9	0.5	0.9	0.9	0.9	0.9
C3	0.4	0.9	1.0	0.9	0.8	0.9	0.9	0.9	0.7
C4	0.9	0.9	0.9	1.0	0.9	0.9	0.9	0.9	0.2
C5	0.9	0.5	0.8	0.9	1.0	0.9	0.9	0.9	0.0
P1	0.5	0.9	0.9	0.9	0.9	1.0	0.9	0.9	0.6
EfficientDet	0.9	0.9	0.9	0.9	0.9	0.9	1.0	0.9	0.1
YOLOv4	0.6	0.9	0.9	0.9	0.9	0.9	0.9	1.0	0.5
RFCN	0.0	0.9	0.7	0.2	0.0	0.6	0.1	0.5	1.0

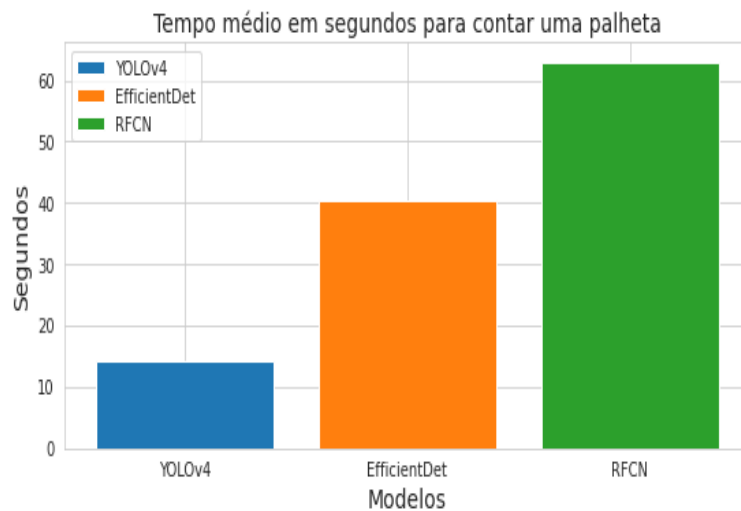
Fonte: Elaborada pelo autor (2021)

média é a mesma para todos os técnicos e modelos. Ou seja, pelo menos um contador tem uma média estatisticamente significativa que apresenta diferença entre as demais médias. Para verificar quais são as diferenças estatísticas entre cada contador individualmente, foi realizado o teste de post-hoc de Nemenyi que mostra a comparação par a par de contadores. A Tabela 7 apresenta esse resultado. Dado $\alpha = 0,05$ os únicos pares de contadores que apresentaram diferença são RFCN X C1 e RFCN X C5 que são as células da tabela cuja fonte é vermelha.

Com a Tabela 7 é possível notar que os modelos EfficientDet e YOLOv4 não apresentam diferenças estatisticamente significativas dos demais contadores, enquanto o modelo RFCN

apresenta discrepância entre duas pessoas. Logo, para este experimento, os modelos EfficientDet e YOLOv4 performaram melhor. Entretanto, algo que também deve ser considerado, é a viabilidade de uso destes modelos em relação ao tempo de contagem de uma palheta. Para isso, foi realizado um experimento com o intuito de saber quanto tempo leva para cada modelo fazer a contagem de uma palheta. O resultado do experimento é apresentado na Figura 33.

Figura 33 – Tempo médio em segundos necessários para cada modelo contar uma palheta. Em azul o tempo do YOLOv4, em laranja o do EfficientDet e em verde o do RFCN.



Fonte: Elaborada pelo autor (2021).

A partir da Figura 33, é possível constatar que a YOLOv4 é a rede mais rápida para contar uma palheta, ou seja, ela leva apenas 14 segundos para fazer a contagem em 204 imagens, enquanto a RFCN é a mais lenta levando mais de um minuto para fazer a contagem. Entretanto, o intervalo de até 2 minutos por palheta se mostraria aceitável, visto que, um ser humano provavelmente levaria mais tempo nesta contagem, pois dependeria do número de ovos e tempo de experiência realizando esta atividade. Além do fator cansaço, a depender do número de palhetas ficará para um técnico fazer a contagem em tempo hábil. Logo, os modelos de máquina ficam em vantagem ao realizar essas atividades, pois para uma máquina não existem essas barreiras, pelo menos a curto prazo.

Desse modo, considerando todos os experimentos apresentados, um modelo de CNN que pode ser escolhido para realizar essa atividade seria o YOLOv4. Além de não ter apresentado diferenças estatísticas significativas entre os técnicos, mostrou-se o mais rápido na realização da tarefa.

6 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho abordou sobre a contagem automática de ovos do mosquito *Ae. aegypti* que é o principal vetor de transmissão de arboviroses como: dengue, Febre Amarela, Zika e Chikungunya que infectam centenas de milhões de pessoas todo ano. A motivação da contagem de ovos de *Ae. aegypti* é o fato dela ser uma forma eficiente de evitar o crescimento no número de casos das arboviroses citadas acima, pois ela serve como um indicador para vigilância vetorial que ajuda na definição de fatores e populações de risco para aplicação de políticas públicas, além de ser um método simples e barato de implantar em ambientes reais.

Entretanto, a desvantagem desse indicador é o fato do processo ser conduzido por técnicos de forma manual, tornando assim o procedimento suscetível a diferentes categorias de erros humanos na contagem, mesmo sendo efetuado por técnicos experientes. Geralmente a contagem de ovos é efetuada com o auxílio de uma lente de aumento ou microscópios ópticos e contadores de mão, tornando-se uma atividade bastante exaustiva até para profissionais mais treinados.

Neste contexto, este trabalho teve como objetivo propor um protótipo de baixo custo para digitalização e contagem automática de ovos de *Ae. aegypti* em palhetas de ovitrampas para eliminar parte do processo manual feito por técnicos. Na realização do trabalho, foram pesquisados e aplicados métodos de inteligência artificial e visão computacional. Entre os métodos aplicados estão: Redes Neurais Convolucionais, Aumento de Dados, Normalização de Histogramas.

Neste trabalho também foi construída uma base de dados com imagens de ovos de *Ae. aegypti* através de imagens capturadas pelo dispositivo digitalizador de palhetas. Para construção da base de dados, este trabalho contou com uma grande contribuição de técnicos da FIOCRUZ-PE para treinamentos onde foram ensinados como identificar os ovos e no fornecimento de palhetas coletadas de ambientes reais.

As contribuições deste trabalho são:

- Criação de uma base de dados com imagens de ovos de *Ae. aegypti*;
- Estudo e aplicação de três modelos de CNNs para contagem de ovos de *Ae. aegypti*;
- Estudo comparativo entre a contagem humana e a contagem com os modelos de CNNs;

- Proposta de um dispositivo de baixo custo que possibilita a digitalização de palhetas de ovitrampas;
- Criação de um protótipo de *software* para que controla o dispositivo de digitalização de palhetas e executa as CNNs para fazer a contagem automática;

Os resultados da pesquisa se mostram satisfatórios para a contagem de palhetas com ovos do mosquito *Ae. aegypti*. Os modelos de CNNs YOLOv4 e o EfficientDet, por exemplo, além de não apresentarem diferenças estatisticamente significativas entre as contagens humanas, conseguem fazer esta tarefa de contar uma palheta em tempos de 14 e 40 segundos, respectivamente. Portanto, fazer a contagem utilizando CNNs como um apoio se mostrou eficiente e pode ajudar profissionais a fazer esta contagem de ovos de maneira mais célere, e, assim, ajudando no controle de problemas de saúde pública causado pelo *Ae. aegypti*.

Os principais pontos de vantagens de utilizar o dispositivo proposto neste trabalho ferramenta na contagem de ovos de *Ae. aegypti* são:

- Melhoria na qualidade de trabalho dos técnicos, pois não precisarão mais gastar tanto tempo fazendo a contagem de ovos;
- Uma menor variância na contagem, visto que, os modelos não apresentam cansaço por esforço repetitivo e podem ser facilmente adaptados para trabalharem em conjunto;
- Agiliza na coleta de dados, pois elimina parte do processo da contagem manual.

Os trabalhos futuros são:

- Aperfeiçoar o equipamento de digitalização das palhetas, atualmente ele só suporta uma palheta por vez, se construir um dispositivo que possa trocar a palheta de forma automática quando uma for finalizada trará mais celeridade ao processo. O ideal seria construir algo como uma pequena esteira que comportasse algumas palhetas.
- Aplicação de filtros diferentes nas imagens ou adquirir uma câmera com foco automático, pois algumas imagens apresentam um desfoque relevante que faz com que os modelos tenham mais dificuldade de detectar.
- Melhoria contínua aplicando novos modelos de CNNs lançados na literatura, visto que, pode ser reaproveitada a base de dados construída no trabalho.

- Por fim, para a construção de uma ferramenta completa será necessário fazer um levantamento de requisitos com os técnicos da FIOCRUZ-PE para entender quais funcionalidades além da contagem, seria interessante para um *software* completo. Para construir esse *software* será reaproveitado o código-fonte do protótipo desenvolvido neste trabalho.

REFERÊNCIAS

- ACIOLI, R. V. et al. *O uso de armadilhas de oviposição (ovitrapas) como ferramenta para monitoramento populacional do Aedes spp em bairros do Recife*. Tese (Doutorado), 2006.
- ALVES, L. D. *Desenvolvimento de um sistema de baixo custo para contagem automática de ovos de Aedes Aegypti utilizando técnicas de processamento de imagem*. Tese (Doutorado), 2016.
- ARDUINO, M. de B.; ÁVILA, G. d. O. de. Aspectos físico-químicos da água de criadouros de aedes aegypti em ambiente urbano e as implicações para o controle da dengue. *Revista de Patologia Tropical/Journal of Tropical Pathology*, v. 44, n. 1, p. 89–100, 2015.
- BAÊTA, K. F. *Avaliação de armadilhas para monitoramento de culicídeos em aeroporto e portos brasileiros*. 2007.
- BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- BONNIN, R. *Machine Learning for Developers: Uplift your regular applications with the power of statistics, analytics, and machine learning*. [S.l.]: Packt Publishing Ltd, 2017.
- BRAGA, A. d. P. *Redes neurais artificiais: teoria e aplicações*. [S.l.]: Livros Técnicos e Científicos, 2000.
- BRAGA, I. A.; VALLE, D. Aedes aegypti: vigilância, monitoramento da resistência e alternativas de controle no brasil. *Epidemiologia e Serviços de Saúde*, Coordenação-Geral de Desenvolvimento da Epidemiologia em Serviços/Secretaria . . . , v. 16, n. 4, p. 295–302, 2007.
- BRASIL, L. M.; GOMES, M. M.; MIOSSO, C. J.; SILVA, M. M. D.; AMVAME-NZE, G. D. Web platform using digital image processing and geographic information system tools: a brazilian case study on dengue. *Biomedical engineering online*, Springer, v. 14, n. 1, p. 1–14, 2015.
- BRASIL, M. da Saúde do. Óbito por arboviroses no brasil, 2008 a 2019. p. 1–4, 2020.
- BRUN, A. L.; MORAES, P. L.; RIZZI, C. B.; RIZZI, R. L. Uma revisão das técnicas computacionais para contagem de ovos de aedes aegypti em imagens de ovitrapas. *Revista Brasileira de Computação Aplicada*, v. 12, n. 3, p. 1–15, 2020.
- CARRASCO, J.; GARCÍA, S.; RUEDA, M.; DAS, S.; HERRERA, F. Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. *Swarm and Evolutionary Computation*, Elsevier, v. 54, p. 100665, 2020.
- CARVALHO, A.; FACELI, K.; LORENA, A.; GAMA, J. *Inteligência Artificial—uma abordagem de aprendizado de máquina*. [S.l.]: LTC, 2011.
- CARVALHO, M. *Esquemas de transferência para aprendizado profundo em classificação de imagens*. Tese (Doutorado) — Dissertação de Mestrado-Universidade Estadual de Campinas, Faculdade de . . . , 2015.

- CHEN, S. W.; SHIVAKUMAR, S. S.; DCUNHA, S.; DAS, J.; OKON, E.; QU, C.; TAYLOR, C. J.; KUMAR, V. Counting apples and oranges with deep learning: A data-driven approach. *IEEE Robotics and Automation Letters*, IEEE, v. 2, n. 2, p. 781–788, 2017.
- COSTA, E. D. *Desenvolvimento de aplicação computacional para detecção e contagem de ovos do aedes aegypti utilizando processamento digital de imagens e lógica fuzzy*. Dissertação (B.S. thesis) — Universidade Federal do Rio Grande do Norte, 2017.
- DAI, J.; LI, Y.; HE, K.; SUN, J. R-fcn: Object detection via region-based fully convolutional networks. *arXiv preprint arXiv:1605.06409*, 2016.
- DONALISIO, M. R.; FREITAS, A. R. R.; ZUBEN, A. P. B. V. Arboviroses emergentes no brasil: desafios para a clínica e implicações para a saúde pública. *Revista de saúde pública*, SciELO Brasil, v. 51, 2017.
- ELPÍDIO, F.; COSTA, L.; ANDRADE, M.; COSTA, E.; BRASIL, L.; RODRIGUES, M. Automatic counting of aedes aegypti eggs deposited in ovitrap by algorithm of digital image processing and artificial neural network. In: IEEE. *2011 Pan American Health Care Exchanges*. [S.l.], 2011. p. 138–139.
- ELPÍDIO, F.; COSTA, L.; PUCCI, G.; ANDRADE, M.; COSTA, E. Automatic identification of aedes aegypti eggs deposited in ovitraps slides using image processing techniques. In: *XXII Brazilian Congress of Biomedical Engineering*. [S.l.: s.n.], 2010.
- FAY, R.; ELIASON, D. A. et al. A preferred oviposition site as a surveillance method for aedes aegypti. *Mosq news*, v. 26, n. 4, p. 531–5, 1966.
- FEITOSA, L. et al. Sistema de contagem automática de ovos do aedes aegypti a partir de processamento de imagens das palhetas de ovitrapas. Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, 2015.
- FIGUEIRÓ, A. C.; SÓTER, A. P.; BRAGA, C.; HARTZ, Z. M. d. A.; SAMICO, I. Análise da lógica de intervenção do programa nacional de controle da dengue. *Revista Brasileira de Saúde Materno Infantil*, SciELO Brasil, v. 10, p. s93–s106, 2010.
- FILHO, O. M.; NETO, H. V. *Processamento digital de imagens*. [S.l.]: Brasport, 1999.
- GABURRO, J.; DUCHEMIN, J.-B.; PARADKAR, P. N.; NAHAVANDI, S.; BHATTI, A. Assessment of icount software, a precise and fast egg counting tool for the mosquito vector aedes aegypti. *Parasites & vectors*, BioMed Central, v. 9, n. 1, p. 1–9, 2016.
- GARCIA, P. S. C.; MARTINS, R.; COELHO, G. L. L. M.; CÁMARA-CHÁVEZ, G. Acquisition of digital images and identification of aedes aegypti mosquito eggs using classification and deep learning. In: IEEE. *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.], 2019. p. 47–53.
- GOLDSCHMIDT, R. R. Uma introdução à inteligência computacional: fundamentos, ferramentas e aplicações. *Rio de Janeiro Brasil: IST-Rio*, v. 1, p. 32, 2010.
- GOMES, A.; SILVA, N.; MARQUES, G.; BRITO, M. Host-feeding patterns of potential human disease vectors in the paraíba valley region, state of são paulo, brazil. *Journal of Vector Ecology: Journal of the Society for Vector Ecology*, v. 28, n. 1, p. 74–78, 2003.

- GONZALEZ, R. C.; WOODS, R. C. *Processamento digital de imagens*. [S.l.]: Pearson Educação, 2010.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. *Deep learning*. [S.l.]: MIT press Cambridge, 2016. v. 1.
- GREGIANINI, T. S.; RANIERI, T.; FAVRETO, C.; NUNES, Z. M. A.; GIANNINI, G. L. T.; SANBERG, N. D.; ROSA, M. T. M. da; VEIGA, A. B. G. da. Emerging arboviruses in rio grande do sul, brazil: Chikungunya and zika outbreaks, 2014–2016. *Reviews in medical virology*, Wiley Online Library, v. 27, n. 6, p. e1943, 2017.
- GUO, G.; ZHANG, N. A survey on deep learning based face recognition. *Computer vision and image understanding*, Elsevier, v. 189, p. 102805, 2019.
- GUSMÃO, G.; MACHADO, S. C.; RODRIGUES, M. A. A new algorithm for segmenting and counting aedes aegypti eggs in ovitraps. In: IEEE. *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. [S.l.], 2009. p. 6714–6717.
- HAYKIN, S. *Redes neurais: princípios e prática*. [S.l.]: Bookman Editora, 2001.
- JUNIOR, C. J. de S.; FIRMO, A. C. A.; OLIVEIRA, R. F. A. P. de; LINS, P. J. B.; LIMA, G. A. de; LIMA, R. A. de. A solution for counting aedes aegypti and aedes albopictus eggs in paddles from ovitraps using deep learning. *IEEE Latin America Transactions*, IEEE, v. 17, n. 12, p. 1987–1994, 2019.
- KITANO, B. T.; MENDES, C. C.; GEUS, A. R.; OLIVEIRA, H. C.; SOUZA, J. R. Corn plant counting using deep learning and uav images. *IEEE Geoscience and Remote Sensing Letters*, IEEE, 2019.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, p. 1097–1105, 2012.
- LETA, S.; BEYENE, T. J.; CLERCQ, E. M. D.; AMENU, K.; KRAEMER, M. U.; REVIE, C. W. Global risk mapping for major diseases transmitted by aedes aegypti and aedes albopictus. *International Journal of Infectious Diseases*, Elsevier, v. 67, p. 25–35, 2018.
- MAINS, J. W.; MERCER, D. R.; DOBSON, S. L. Digital image analysis to estimate numbers of aedes eggs oviposited in containers. *Journal of the American Mosquito Control Association*, NIH Public Access, v. 24, n. 4, p. 496, 2008.
- MCCARTHY, J. *What is artificial intelligence?* 1998.
- MELLO, C. A.; SANTOS, W. P. D.; RODRIGUES, M. A.; CANDEIAS, A. L. B.; GUSMAO, C. M. Image segmentation of ovitraps for automatic counting of aedes aegypti eggs. In: IEEE. *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. [S.l.], 2008. p. 3103–3106.
- MITCHELL, T. *Machine Learning*. [S.l.]: McGraw-Hill Higher Education, 1997.
- MORAIS, H.; SANTOS, O.; ROCHA, M.; ALMEIDA, T.; BRASIL, L.; AMVAME-NZE, G.; MIOSSO, C.; COSTA, M.; PIZO, G. Monitoring information system of aedes aegypti reproduction. In: SPRINGER. *World Congress on Medical Physics and Biomedical Engineering, June 7-12, 2015, Toronto, Canada*. [S.l.], 2015. p. 1423–1426.

- PADILLA, R.; NETTO, S. L.; SILVA, E. A. da. A survey on performance metrics for object-detection algorithms. In: IEEE. *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*. [S.l.], 2020. p. 237–242.
- PARKER, J. R. *Algorithms for image processing and computer vision*. [S.l.]: John Wiley & Sons, 2010.
- PEREIRA, D. G.; AFONSO, A.; MEDEIROS, F. M. Overview of friedman's test and post-hoc analysis. *Communications in Statistics-Simulation and Computation*, Taylor & Francis, v. 44, n. 10, p. 2636–2653, 2015.
- PORTELA, M. Contagem automática de ovos de aedes aegypti em imagens de ovitrampas. *Escola Politécnica de Pernambuco*, 2009.
- POSSATTI, L. C.; GUIDOLINI, R.; CARDOSO, V. B.; BERRIEL, R. F.; PAIXÃO, T. M.; BADUE, C.; SOUZA, A. F. D.; OLIVEIRA-SANTOS, T. Traffic light recognition using deep learning and prior maps for autonomous cars. In: IEEE. *2019 international joint conference on neural networks (IJCNN)*. [S.l.], 2019. p. 1–8.
- RIZZI, C. B.; BRUN, A. L.; GALANTE, G.; RIZZI, R. L. Sigdengue: Um sistema de informação para o acompanhamento e gestão de ações sobre dengue com enfoque às atividades de notificação, raio e bloqueio. *iSys-Revista Brasileira de Sistemas de Informação*, v. 9, n. 1, p. 101–117, 2016.
- ROCHA, C. D. F. da; BIZERRA, A. M. C.; MAGALHÃES, D. A. Contagem automática de ovos do aedes aegypti em palhetas de ovitrampas: Um sistema para aquisição e processamento de imagens. *CONTAGEM AUTOMÁTICA DE OVOS DO AEDES AEGYPTI EM PALHETAS DE OVITRAMPAS: UM SISTEMA PARA AQUISIÇÃO E PROCESSAMENTO DE IMAGENS*, Atena Editora, p. 1–388, 2019.
- RUSSELL, S.; NORVIG, P. *Artificial intelligence: a modern approach*. 2002.
- SANTOS, E. P. dos. Ovitrampas: Alternativa sustentável para o monitoramento do aedes aegypti e ferramenta para divulgação científica. *RECIMA21-Revista Científica Multidisciplinar-ISSN 2675-6218*, v. 1, n. 1, 2021.
- SANTOS, W.; MELLO, C.; MESQUITA, R.; RODRIGUES, M.; CANDEIAS, A.; GUSMÃO, C. Um algoritmo para contagem automática de ovos do mosquito aedes aegypti em ovitrampas para controle da dengue. In: *CBE2008: Anais do 21º Congresso Brasileiro de Engenharia Biomédica*. [S.l.: s.n.], 2008.
- SEBASTIANI, R. G.; VIALI, L. Teste de hipóteses: uma análise dos erros cometidos por alunos de engenharia. *Bolema-Boletim de Educação Matemática*, v. 24, n. 40, p. 835–854, 2011.
- SILVA, M. G. N. M. d.; RODRIGUES, M. A. B.; ARAUJO, R. E. d. Sistema de aquisição e processamento de imagens de ovitrampas para o combate a dengue. *Revista Brasileira de engenharia Biomédica*, SciELO Brasil, v. 28, n. 4, p. 364–374, 2012.
- SILVA, M. G. N. M. da; RODRIGUES, M. A. B.; ARAUJO, R. D. Aedes aegypti egg counting system. In: IEEE. *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. [S.l.], 2011. p. 6810–6812.

SILVA, M. G. N. M. R. M. A. B.; E., A. R. Um novo método de segmentação e contagem de ovos do aedes aegypti em ovitrampas. In: *XXII Congresso Brasileiro de Engenharia Biomédica (CBEB)*. [S.l.: s.n.], 2010. p. 861–865.

SOGHIGIAN, J.; GLORIA-SORIA, A.; ROBERT, V.; GOFF, G. L.; FAILLOUX, A.-B.; POWELL, J. R. Genetic evidence for the origin of aedes aegypti, the yellow fever mosquito, in the southwestern indian ocean. *Molecular Ecology*, Wiley Online Library, v. 29, n. 19, p. 3593–3606, 2020.

SOKOLOVA, M.; JAPKOWICZ, N.; SZPAKOWICZ, S. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In: SPRINGER. *Australasian joint conference on artificial intelligence*. [S.l.], 2006. p. 1015–1021.

SOUZA, L. J. de. *Dengue, Zika e Chikungunya–Diagnóstico, Tratamento e Prevenção*. [S.l.]: Editora Rubio, 2016.

TAN, M.; PANG, R.; LE, Q. V. Efficientdet: Scalable and efficient object detection. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2020. p. 10781–10790.

TAVEIRA, L. A.; FONTES, L. R.; NATAL, D.; GOMES, A. d. C. Manual de diretrizes e procedimentos no controle do aedes aegypti. In: *Manual de diretrizes e procedimentos no controle do Aedes aegypti*. [S.l.: s.n.], 2001. p. 108–108.

TEICH, V.; ARINELLI, R.; FAHHAM, L. Aedes aegypti e sociedade: o impacto econômico das arboviroses no brasil. *JBES: Brazilian Journal of Health Economics/Jornal Brasileiro de Economia da Saúde*, v. 9, n. 3, 2017.

TIBA, A.; ARAUJO, H. A. Sistema autonomo de reconhecimento e contagem de ovos de aedes aegypti. *Sociedade Brasileira de Inteligencia Computacional (SBIC)*, 2011.

VARGAS, A. C. G.; PAES, A.; VASCONCELOS, C. N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: *Proceedings of the xxix conference on graphics, patterns and images*. [S.l.: s.n.], 2016. v. 1, n. 4.

WILKE, A. B. B.; VASQUEZ, C.; CARVAJAL, A.; MEDINA, J.; CHASE, C.; CARDENAS, G.; MUTEBI, J.-P.; PETRIE, W. D.; BEIER, J. C. Proliferation of aedes aegypti in urban environments mediated by the availability of key aquatic habitats. *Scientific reports*, Nature Publishing Group, v. 10, n. 1, p. 1–10, 2020.

YUSSOF, W. N. J. H. W.; MAN, M.; HITAM, M. S.; HAMID, A. A. K. A.; AWALLUDIN, E. A.; BAKAR, W. A. W. A. Wavelet-based auto-counting tool of aedes eggs. In: *Proceedings of the 2018 International Conference on Sensors, Signal and Image Processing*. [S.l.: s.n.], 2018. p. 56–59.