



**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO  
UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO**



**WILLIAN RAMON BARBOSA BESSA**

**SOLUÇÕES BASEADAS EM REDES NEURAIS  
CONVOLUCIONAIS PARA CONTAGEM AUTOMÁTICA DE  
ANIMAIS AQUÁTICOS**

**MOSSORÓ**

**2022**

**WILLIAN RAMON BARBOSA BESSA**

**SOLUÇÕES BASEADAS EM REDES NEURAIIS  
CONVOLUCIONAIS PARA CONTAGEM AUTOMÁTICA DE  
ANIMAIS AQUÁTICOS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação - associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido, para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Francisco Milton Mendes Neto - UFERSA

Coorientador: Prof. Dr. Mário Wedney de Lima Moreira- IFCE

**MOSSORÓ**

**2022**

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

BB557 Bessa, Willian Ramon Barbosa.  
s Soluções baseadas em redes neurais  
convolucionais para contagem automática de  
animais aquáticos / Willian Ramon Barbosa Bessa. -  
2022.  
77 f. : il.

Orientador: Francisco Milton Mendes Neto.  
Coorientador: Mário Wedney de Lima Moreira.  
Dissertação (Mestrado) - Universidade Federal  
Rural do Semi-árido, Programa de Pós-graduação em  
Ciência da Computação, 2022.

1. Aquicultura. 2. Redes neurais artificiais.  
3. visão computacional. 4. conjunto de dados. 5.  
limiar de ativação . I. Neto, Francisco Milton  
Mendes, orient. II. Moreira, Mário Wedney de  
Lima, co-orient. III. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade  
com AACR2 e os dados fornecidos pelo(a) autor(a).  
Biblioteca Campus Mossoró / Setor de Informação e Referência  
Bibliotecária: Keina Cristina Santos Sousa e Silva  
CRB: 15/120

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

**WILLIAN RAMON BARBOSA BESSA**

**SOLUÇÕES BASEADAS EM REDES NEURAIS  
CONVOLUCIONAIS PARA CONTAGEM AUTOMÁTICA DE  
ANIMAIS AQUÁTICOS**

Dissertação apresentada ao Programa de  
Pós-Graduação em Ciência da Computação para  
a obtenção do título de Mestre em Ciência da  
Computação.

Aprovada em:

**BANCA EXAMINADORA**

---

Prof. Dr. Francisco Milton Mendes Neto - UFERSA  
Presidente

---

Prof. Dr. Mário Wedney de Lima Moreira- IFCE  
Primeiro Membro

---

Prof. Dr. Carlos Heitor Pereira Liberalino - UERN  
Segundo Membro

---

Prof. Dr. Gabriel Antoine Louis Paillard - UFC  
Terceiro Membro



À minha família, por me incentivam a ir mais longe, e a todos os meus professores, que tornaram isso possível.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus pela graça da vida e por ter me sustentado em todos os momentos.

À toda a minha família, em especial ao meu avô Manoel Loureiro, por sempre me lembrar da importância do compromisso com o estudo e o trabalho. A minha esposa Adrielle, que sempre me incentivou. Aos meus filhos Isaac e Alice, as razões da minha vida e meus maiores projetos.

Agradeço ao professor Damian Keller, por me introduzir ao universo da pesquisa científica, por me guiar em diversos projetos e pelo incentivo a ingressar nessa carreira tão gratificante que é a docência. Obrigado também a todo o grupo do Núcleo Amazônico de Pesquisa Musical (NAP), pela parceria e companheirismo ao longo de tantos anos.

Agradeço aos professores, coordenadores e toda equipe do Programa de Pós-graduação em Ciência da Computação (PPgCC), pela dedicação, zelo e empenho constante em aprimorar o programa. Agradeço ao professor Carlos Heitor Liberalino pela excelente recepção.

Agradeço aos meus orientadores Milton Mendes e Mario Wedney, por acreditarem no meu potencial, pela disponibilidade em todos os momentos, pela paciência e por darem todo apoio necessário para conclusão do curso.

Agradeço a toda equipe da Crevettic, e aos colaboradores do Instituto Federal do Ceará (IFCE) - Aracati, por sempre estarem dispostos a ajudar e contribuir com o trabalho. Sem vocês, não teria sido possível realizá-lo.

A todos os amigos de turma que fiz durante o decorrer do mestrado, mesmo estando geograficamente distante, sempre me senti acolhido.

Ao Conselho Nacional de Pesquisa (CNPq) pelo apoio financeiro para realização deste trabalho de pesquisa.

A todos que fizeram parte de forma direta ou indireta a este projeto, muito obrigado!

## **EPÍGRAFE**

"Ainda que eu falasse as línguas dos homens e dos anjos, e não tivesse amor, seria como o metal que soa ou como o sino que tine."

(1 Coríntios 13:1)

## RESUMO

A aquicultura é o processo de cultivo de organismos com o habitat predominantemente aquático, sendo hoje uma importante atividade na produção de alimento humano. Apesar de sua importância, ainda existe uma série de atividades que são realizadas quase que exclusivamente de forma manual, dentre elas, pode-se destacar a contagem de animais. Logo, esse trabalho apresenta a proposição de um conjunto de soluções para dar suporte à atividade de contagem de animais aquáticos, utilizando técnicas de visão computacional e de aprendizado de máquina, através de aprendizado profundo, com o diferencial de que o usuário final será capaz de acessar as soluções via *smartphone*. Como a solução final será disponibilizada para o usuário utilizar de forma autônoma, foi desenvolvido um modelo para ser usado como limiar de ativação, que avalia se a imagem enviada está em conformidade ou não. Atualmente o modelo está com 99% de precisão. Foi desenvolvido também um modelo de contagem baseado em *YOLOv4* que alcançou 98.50% de *mAP* e 98.70% de acurácia, obtendo assim um excelente resultado.

**Palavras-chave:** Aquicultura, redes neurais artificiais, visão computacional, conjunto de dados, limiar de ativação.

## ABSTRACT

Aquaculture is the process of cultivating organisms from predominantly aquatic habitat, and it is today an important activity for the production of human food. Despite its importance, there are still a number of activities that are presented almost exclusively manually, among them, the process of counting animals can be highlighted. Therefore, this work presents a proposal for support solutions to the activity of counting aquatic animals, using computer vision and machine learning techniques, through deep learning algorithms, with the particularity that the end-user will be able to access the solutions via smartphone. Since the final solution will be provided anonymously to the user, it was then built a test model to be used as an activation threshold, evaluating if the image that was sent to the system is in accordance or not. Currently said model has a 99% precision rate. A counting model based on *YOLOv4* was also developed, which reached 98.5% of *mAP* and 98.70% of accuracy, thus obtaining an excellent result.

**Keywords:** Aquaculture, Artificial Neural Networks, Computer Bision, Dataset, Threshold.

## LISTA DE FIGURAS

Figura 1 – Peneiras utilizadas em fazenda de criação de peixes. . . . .	20
Figura 2 – Esquema de um sistema de visão computacional . . . . .	23
Figura 3 – Exemplo de uma Rede Neural Artificial Multicamadas (Deep Learning) . . . . .	24
Figura 4 – Ilustração de um modelo de aprendizado profundo . . . . .	25
Figura 5 – Representação de uma entrada de imagem RGB . . . . .	26
Figura 6 – Exemplo de Overfitting . . . . .	27
Figura 7 – Ferramenta Trello . . . . .	35
Figura 8 – Padrão utilizado para nomear as imagens. . . . .	38
Figura 9 – Exemplos de imagens em não conformidade e conformidade. . . . .	40
Figura 10 – Apresentação do apoio construído em impressora 3D. . . . .	41
Figura 11 – Exemplo de imagem com caixas delimitadoras no programa <i>LabelImg</i> . . . . .	42
Figura 12 – Gráfico de imagens pré-processadas e as que estão aguardando processamento por quantidade de animais. . . . .	43
Figura 13 – Gráfico de imagens pré-processadas e aguardando por subconjuntos. . . . .	43
Figura 14 – Diagrama de atividade para análise de conformidade. . . . .	45
Figura 15 – Exemplos de aplicação de aumento de dados usando a API Keras. . . . .	46
Figura 16 – Arquitetura geral da solução . . . . .	50
Figura 17 – MIC - Exibição da tela Swagger . . . . .	50
Figura 18 – Proposta de nova tela de povoamento . . . . .	51
Figura 19 – Matriz Confusão - Modelos. . . . .	53
Figura 20 – Gráfico apresentando o <i>mAP</i> do modelo de treinamento das redes <i>YOLOv4</i> e <i>YOLOv4-tiny</i> da primeira geração, com as configurações informadas na seção 5.3 . . . . .	56
Figura 21 – Gráfico apresentando a <i>loss</i> do modelo de treinamento das redes <i>YOLOv4</i> e <i>YOLOv4-tiny</i> da primeira geração, com as configurações informadas na seção 5.3 . . . . .	57
Figura 22 – Gráfico apresenta a relação entre a quantidade de caixas delimitadoras e quantidade de animais previstos do modelo <i>YOLOv4</i> primeira geração. . . . .	58
Figura 23 – <i>Absolute Error</i> - Conjunto de Validação <i>YOLOv4</i> primeira geração. . . . .	59
Figura 24 – Imagens com 58 animais dos subconjuntos <i>tilapia_002</i> e <i>tilapia_003</i> . . . . .	60

Figura 25 – Imagens com 58 animais dos subconjuntos tilapia_002 e tilapia_003 retornadas pelo modelo de contagem <i>YOLOv4</i> primeira geração. . . .	60
Figura 26 – Gráficos apresentando <i>loss</i> e <i>mAP</i> do modelo de treinamento das redes <i>YOLOv4</i> e <i>YOLOv4-tiny</i> segunda geração, com as configurações informadas na seção 5.3 . . . . .	62
Figura 27 – Gráfico apresentando a relação entre a quantidade de caixas delimitadoras e quantidade de animais previstos do modelo <i>YOLOv4</i> da segunda geração. . . . .	63
Figura 28 – <i>Absolute Error</i> - Conjunto de Validação <i>YOLOv4</i> da segunda geração.	64
Figura 29 – Imagens processadas <i>YOLOv4</i> da primeira geração. . . . .	64

## LISTA DE TABELAS

Tabela 1 – Compilado - Trabalhos sobre automação de contagem. . . . .	32
Tabela 2 – Dados da planilha info_vivarium_dataset. . . . .	37
Tabela 3 – Parâmetros utilizados na configuração das redes e na geração de novas imagens . . . . .	46
Tabela 4 – Resultados obtidos. . . . .	53
Tabela 5 – Resultados dos modelos de contagem da primeira geração. . . . .	57
Tabela 6 – Resultados dos modelos de contagem da segunda geração. . . . .	62
Tabela 7 – Resultados dos modelos de contagem da segunda geração para tilapia_005	65



## LISTA DE ABREVIATURAS E SIGLAS

ABCC	<i>Associação Brasileira de Criadores de Camarão no Brasil</i>
AM	<i>Aprendizado de máquina</i>
AP	<i>Average Precision</i>
CNN	<i>Redes Neurais Convolucionais</i>
FAO	<i>Food and Agriculture Organization of the United Nations</i>
IA	<i>Inteligência Artificial</i>
IoU	<i>Intersection over Union</i>
mAP	<i>Mean Average Precision</i>
MAPE	<i>Mean Absolute Percentage Error</i>
RMSE	<i>Root Mean Squared Error</i>
RNAs	<i>Redes Neurais Artificiais</i>
YOLO	<i>You Only Look Once</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>14</b>
<b>1.1</b>	<b>Motivação . . . . .</b>	<b>15</b>
<b>1.2</b>	<b>Problemática . . . . .</b>	<b>16</b>
<b>1.3</b>	<b>Objetivo . . . . .</b>	<b>16</b>
<b>1.3.1</b>	<b><i>Objetivos específicos . . . . .</i></b>	<b>16</b>
<b>1.4</b>	<b>Organização do documento . . . . .</b>	<b>17</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>18</b>
<b>2.1</b>	<b>Aquicultura . . . . .</b>	<b>18</b>
<b>2.1.1</b>	<b><i>Piscicultura . . . . .</i></b>	<b>18</b>
<b>2.1.2</b>	<b><i>Carcinicultura . . . . .</i></b>	<b>19</b>
<b>2.1.3</b>	<b><i>Contagem manual de alevinos e pós-larvas . . . . .</i></b>	<b>19</b>
<b>2.2</b>	<b>Inteligência Artificial . . . . .</b>	<b>21</b>
<b>2.2.1</b>	<b><i>Aprendizado de Máquina . . . . .</i></b>	<b>21</b>
<b>2.2.2</b>	<b><i>Visão Computacional . . . . .</i></b>	<b>21</b>
<b>2.2.3</b>	<b><i>Redes Neurais Artificiais . . . . .</i></b>	<b>23</b>
<b>2.2.4</b>	<b><i>Aprendizado Profundo . . . . .</i></b>	<b>24</b>
<b>2.2.5</b>	<b><i>Redes Neurais Convolucionais . . . . .</i></b>	<b>25</b>
<b>2.2.6</b>	<b><i>Overfitting . . . . .</i></b>	<b>26</b>
<b>3</b>	<b>TRABALHOS RELACIONADOS . . . . .</b>	<b>28</b>
<b>3.1</b>	<b>Detecção de outliers . . . . .</b>	<b>28</b>
<b>3.2</b>	<b>Contagem de animais . . . . .</b>	<b>29</b>
<b>4</b>	<b>METODOLOGIA . . . . .</b>	<b>33</b>
<b>4.1</b>	<b>SCRUM . . . . .</b>	<b>34</b>
<b>5</b>	<b>SOLUÇÕES . . . . .</b>	<b>36</b>
<b>5.1</b>	<b><i>Design Vivarium . . . . .</i></b>	<b>36</b>
<b>5.1.1</b>	<b><i>Organização . . . . .</i></b>	<b>37</b>
<b>5.1.2</b>	<b><i>Especificações de captura de imagens . . . . .</i></b>	<b>38</b>
<b>5.1.3</b>	<b><i>Subconjuntos . . . . .</i></b>	<b>41</b>
<b>5.1.4</b>	<b><i>Pré-processamento . . . . .</i></b>	<b>41</b>
<b>5.2</b>	<b>Modelo de Avaliação de Imagens Vivarium - MAIV . . . . .</b>	<b>44</b>

5.3	Modelos de contagem . . . . .	47
5.4	Disponibilização dos Modelos . . . . .	49
5.4.1	<i>Arquitetura</i> . . . . .	49
6	<b>RESULTADOS</b> . . . . .	52
6.1	<b>MAIV</b> . . . . .	52
6.2	<b>Modelos de Contagem</b> . . . . .	54
6.2.1	<i>Métricas</i> . . . . .	54
6.2.2	<i>Gerações</i> . . . . .	55
6.2.3	<i>Primeira Geração de Modelos</i> . . . . .	55
6.2.4	<i>Segunda Geração de Modelos</i> . . . . .	61
7	<b>CONCLUSÃO</b> . . . . .	66
8	<b>PRODUTIVIDADE ACADÊMICA E CONQUISTAS</b> . . . . .	67
8.1	<b>Trabalhos Publicados</b> . . . . .	67
8.2	<b>Trabalhos apresentados</b> . . . . .	67
8.3	<b>Capítulo de livros</b> . . . . .	68
	<b>REFERÊNCIAS</b> . . . . .	69

## 1 INTRODUÇÃO

A aquicultura pode ser definida como o cultivo de organismos (animais ou plantas) que tenham na água doce, marinha ou salobra o seu principal *habitat* (POUEY; CAMARGO, 2005). De acordo com o levantamento feito pela *Food and Agriculture Organization of the United Nations* (FAO), em 2018 a produção de produtos oriundos da aquicultura foi de 114,5 milhões de toneladas, atingindo um novo recorde, sendo 82,1 milhões de toneladas de animais aquáticos e 32,4 milhões de toneladas de algas aquáticas (FAO, 2020). Dentre os diversos ramos que existem na aquicultura, pode-se destacar a carcinicultura e a piscicultura, que são respectivamente áreas responsáveis pelo estudo e desenvolvimento de técnicas de cultivo e reprodução de camarões e peixes.

A carcinicultura é um importante setor na economia do Nordeste, principalmente do Ceará e Rio Grande do Norte, maiores produtores nacionais. Segundo a *Associação Brasileira de Criadores de Camarão no Brasil* (ABCC), a produção de camarão no Brasil encontra-se em crescimento constante. Em 2019 foram produzidas 90.000 mil toneladas do alimento em cativeiro. Desse total, Ceará e Rio Grande do Norte produziram respectivamente 38% e 28% da produção nacional<sup>1</sup>. Um mercado que em 2019 movimentou aproximadamente R\$1.18 bilhões de reais em valores nominais (XIMENES, 2021).

A produção brasileira de peixes de cultivo atingiu a marca de 802.930 toneladas em 2020, representando um crescimento de 5,93% em relação ao ano anterior e de 38% em relação à produção de 2014 que foi de 578.800 toneladas<sup>2</sup>. Esses dados corroboram a afirmação de que a piscicultura brasileira vem se desenvolvendo de maneira robusta nos últimos anos, com significativos avanços em termos de aumento da produção e profissionalização do setor (FILHO *et al.*, 2020).

O Brasil possui um grande potencial para o desenvolvimento de diversas modalidades de aquicultura, com os requisitos hidrográficos para se tornar um grande produtor, com capacidade para abastecer seu mercado interno e poder competir em nível mundial nas exportações. Porém, apesar dos constantes resultados positivos, existem uma série de desafios a serem enfrentados, como a dificuldade de escoamento e o elevado custo de produção, sendo que esses fatores podem ser mitigados através de investimentos permanentes em tecnologias de manejo, logística e biossegurança (XIMENES, 2021).

<sup>2</sup> Informações extraídas de (ABCC, 2021)

<sup>2</sup> Informações extraídas de (Peixe BR, 2021)

Em um mercado em constante ascensão como o da aquicultura, se faz necessário uma evolução acelerada não somente dos meios de produção, mas também da modernização das atividades realizadas durante todo o processo. Portanto, é necessário que se desenvolvam tecnologias que possam dar suporte às atividades e atender as constantes demandas do mercado. Dentre as diversas soluções computacionais que podem ser aplicadas ao longo de toda cadeia produtiva, pode-se destacar o uso de *Inteligência Artificial* (IA).

## 1.1 Motivação

Assim como em outros setores de criação de animais, a aquicultura tem como objetivo a criação e manutenção de organismos, durante um certo período de tempo em uma determinada área, sendo que a não observância dos limites recomendados de povoamento podem acarretar prejuízos aos produtores. Quando a densidade de estocagem é maior que o ideal, ela pode acarretar em uma série de problemas para o produtor, dentre eles: baixo nível de crescimento, morte, canibalismo e até alterações na qualidade da água (KRUMMENAUER *et al.*, 2011). Já a densidade de estocagem baixa acarreta em um subaproveitamento da área destinada ao cultivo de animais, podendo não gerar a rentabilidade esperada.

A densidade de estocagem está diretamente relacionada à quantidade de animais que são inseridos nos meios de criação, logo a contagem desses animais no momento de iniciação do cultivo é um processo determinante no sucesso da produção. A contagem de animais na aquicultura é um problema real e uma necessidade prioritária, devendo ser realizada de forma rápida, confiável e eficiente, pois impacta diretamente no lucro do produtor, visto que é a partir da quantidade de animais que se dimensiona diversas variáveis como: quantidade de alimento, qualidade da água, taxa de crescimento e saúde dos animais (LI *et al.*, 2021).

A maioria dos aquicultores adotam formas manuais de contagem, o que demanda um trabalho humano intensivo e altamente demorado, não havendo garantia sobre a qualidade do processo de contagem, uma vez que as medidas para avaliar o desempenho, como o ato de recontar, seriam inviáveis e demoradas. Existem algumas soluções propostas para automatização e suporte da atividade como (SALAZAR *et al.*, 2015; HERNÁNDEZ-ONTIVEROS *et al.*, 2018; LABUGUEN *et al.*, 2012), que são mediadas por diversos equipamentos, porém essas alternativas são onerosas, necessitam de equipamentos espe-

cíficos e de uma equipe técnica especializada, o que dificulta o acesso dos produtores de pequeno e médio porte.

## 1.2 Problemática

A contagem de animais na aquicultura é uma atividade que ocorre em várias etapas do processo de cultivo, sobretudo nas etapas iniciais, e durante a comercialização das pós-larvas e alevinos, devido ao fato de que, nessa etapa, os animais são vendidos por unidade. A aquicultura no Brasil é normalmente dividida em duas fases distintas, sendo a primeira a de produção das larvas, que são realizadas em fazendas de reprodução e larvicultura; e a segunda é a engorda realizada em fazendas de engorda (GOMES *et al.*, 2003).

Para as fazendas de médio e pequeno porte, normalmente trabalham em apenas uma dessas atividades, logo as fazendas de reprodução não trabalham com a engorda, do mesmo modo que as de engorda não trabalham com a produção de larvas. Conforme dados e instruções do Serviço Nacional de Aprendizagem Rural (SENAR, 2017a; SENAR, 2017b), no processo de aquisição de animais para engorda é uma prática comum a realização de duas contagens, uma por parte do vendedor no momento do despacho e uma por parte do comprador antes de liberar os animais nos tanques de engorda.

## 1.3 Objetivo

Este projeto tem como objetivo geral realizar a criação de mecanismos capazes de auxiliar o processo de contagem de alevinos e pós-larvas, através do desenvolvimento de modelos de predição utilizando técnicas de aprendizagem profunda (*deep learning*) e visão computacional. Garantindo assim maior segurança, agilidade, precisão e consequentemente melhorando a produtividade das fazendas.

### 1.3.1 Objetivos específicos

- Analisar o estado da arte sobre modelos de predição de contagens voltados para aquicultura;
- Desenvolver e aplicar uma metodologia para criação de subconjuntos de imagens de animais aquáticos em processo de contagem;

- Levantar os requisitos, para criar modelos de contagem e de detecção de *outliers*;
- Desenvolver uma plataforma para disponibilização dos modelos de contagem e de detecção de *outliers*;
- Avaliar o desempenho dos modelos propostos.

#### **1.4 Organização do documento**

Este trabalho está organizado conforme a seguinte estrutura: inicia-se com essa introdução, no Capítulo 2, é apresentada a fundamentação teórica, contendo uma revisão sobre os principais assuntos abordados neste trabalho. O capítulo 3 apresenta trabalhos relacionados à remoção de *outliers* em sistemas de visão computacional e de contagem de animais aquáticos. O capítulo 4 configura-se pelo detalhamento da metodologia utilizada. O capítulo 5 apresenta todo o processo de desenvolvimento das soluções propostas e o capítulo 6 expõe os resultados obtidos. O capítulo 7, que contém as considerações finais, bem como são apresentadas as propostas para trabalhos futuros. Por fim, tem-se o capítulo 8 onde é apresentado a produtividade acadêmica desenvolvida no decorrer do projeto.

## 2 FUNDAMENTAÇÃO TEÓRICA

Para melhor entendimento da proposta, convém apresentar, o embasamento teórico sobre os principais conceitos que se articulam no decorrer do trabalho.

### 2.1 Aquicultura

Conforme Rodrigues *et al.* (2012), é interessante diferenciar primeiramente o conceito de pesca e aquicultura, pois ambos podem ser realizados no mar ou em águas continentais, porém a pesca é a retirada de organismos aquáticos da natureza sem prévio cultivo, podendo ocorrer em escala industrial ou artesanal, enquanto a aquicultura é o processo de produção em cativeiro de organismos com *habitat* predominantemente aquático, tais como peixes, camarões, rãs, algas, entre outras espécies.

O cultivo de organismos aquáticos pelo ser humano é uma atividade que se iniciou na China, há aproximadamente 4.000 anos por meio da criação de peixes e de macroalgas marinhas, com o objetivo de serem utilizadas como fonte de alimentos. De acordo com Américo *et al.* (2013), a exploração predatória dos estoques naturais, bem como a crescente diferença entre a quantidade de pescado capturado e a demanda de consumo, tornaram a aquicultura um das alternativas mais viáveis no mundo para produção de alimento direcionado ao consumo humano de alto valor proteico.

#### 2.1.1 Piscicultura

A piscicultura é um ramo da aquicultura especializada na criação de peixes, e apesar das incertezas associadas a pandemia no ano de 2020, no Brasil houve uma produção de 802.930 toneladas, o que corresponde a um crescimento de 5,93% em relação ao ano anterior. Essa produção, no ano de 2020, estava dividida da seguinte maneira: 486.155 toneladas de tilápias, representando 60,6% da produção; seguida por peixes nativos, com 278.671 toneladas e outras espécies (carpa, truta, pangasius, principalmente), com 38.104 toneladas (Peixe BR, 2021).



### 2.1.2 Carcinicultura

A carcinicultura é a área especializada na criação camarões, podendo ser de espécies marinhas ou de água doce. No Brasil, há a predominância da produção da espécie marinha *Litopenaeus Vannamei*, que é conhecida popularmente como camarão cinza. Apesar do cenário promissor, os produtores vem enfrentando uma série de dificuldades, dentre elas o risco de contaminação dos seus cultivos pelo vírus da mionecrose infecciosa (*Infectious Myonecrosis Virus*), bem como o vírus da mancha branca (*White Spot Syndrome Virus*), e, em virtude disto está sendo realizado um acompanhamento mais efetivo do manejo e saúde dos camarões (BRAGA, 2020).

### 2.1.3 Contagem manual de alevinos e pós-larvas

Existe uma série de técnicas de contagem de alevinos e pós-larvas, a depender da quantidade que será contada e da preferência do produtor. A seguir, é descrito um conjunto de métodos encontrados na revisão literária (SALAZAR *et al.*, 2015; SENAR, 2017a; SENAR, 2017b; KHANTUWAN; KHIRIPET, 2012):

- Contagem manual, quando uma pessoa ou mais é responsável, realizando a contagem de um por um, transferindo os animais de um container para outro. Este método tem como desvantagem a velocidade e está sujeito a erros humanos, já que é uma atividade monótona e repetitiva;
- Por volume seco é um método que se constitui quando há um local onde os alevinos e as pós-larvas estejam armazenados e já se sabe o volume dele, possibilitando a retirada de uma ou mais amostras, com a ajuda de um recipiente, e então é realizada a contagem dos seres. A partir do valor do volume e da contagem das amostras, é realizada uma estimativa da quantidade de animais no volume total. Esse método carece de precisão, pois é afetado pelos diferentes tamanhos que um lote de animais pode ter;
- Por volume com água é um método semelhante a anterior, porém essa estimativa é realizada com o volume dos animais juntamente com água. Ele apresenta uma desvantagem semelhante ao método anterior, acrescido da dificuldade de coleta de amostras que representem a distribuição igualitária do volume total, tendo em vista que os animais estão dispersos na água, no local em que estão armazenados;

- Por comparação visual é um processo que consiste na determinação do número de animais por meio da comparação entre dois recipientes, sendo que um deles já se sabe a quantidade que está armazenada. Esse método apresenta um alto índice de erro, pois a determinação é subjetiva;
- Com um filtro ou peneira em que é conhecido o número que peixes que podem ser armazenados, onde a contagem é feita à medida que são movidos os animais de um local para outro com o instrumento, a exemplo de peneiras, como a apresentada na Figura 1;
- Método gravimétrico se constitui quando há a ajuda de uma balança eletrônica para encontrar uma relação entre o peso e a quantidade de animais. Esse método apresenta um fator dificultador que é o fato de nem todos os filhotes possuírem o mesmo tamanho e peso.

Figura 1 – Peneiras utilizadas em fazenda de criação de peixes.



Fonte: autoria própria (2021).

A contagem manual pode ser considerada a mais confiável, porém possui desvantagens significativas em relação a manipulação física dos animais e ao tempo de execução da atividade, o que pode acarretar em erros devido ao processo de exaustão e de descontração ocasionados ao executar a atividade por um longo período de tempo. Dos métodos anteriormente expostos, exceto a contagem manual, todos compartilham da desvantagem da redução da precisão em determinar o número correto de animais devido à falta de

homogeneidade em relação ao tamanho e ao peso dos animais (SALAZAR *et al.*, 2015). Vale ressaltar, também, que os métodos por estimativa podem ter, em alguma etapa, a contagem manual das amostras.

## **2.2 Inteligência Artificial**

Segundo Russell e Norvig (2002), IA é um das áreas mais recentes das ciências, cujo o objetivo não é apenas compreender, mas também construir entidades inteligentes que funcionem de forma autônoma em ambientes complexos e mutáveis, sendo dividido em diversos campos de atuação. Nesse contexto, pode-se entender entidades inteligentes, como sistemas computacionais capazes de, a partir de dados apresentados, aprender e fazer inferências acerca de problemas propostos.

A IA pode ainda ser definida como a ciência e engenharia de fazer máquinas inteligentes, em especial programas de computador inteligentes, baseando-se ou não em métodos biologicamente observáveis (MCCARTHY, 2007). Essa prática está em constante processo de evolução e existem diversos algoritmos que são utilizados, dentre eles pode-se destacar os tipos: supervisionados, não supervisionados, por reforço e semi-supervisionados (TAN *et al.*, 2009).

### **2.2.1 *Aprendizado de Máquina***

*Aprendizado de máquina* (AM) surgiu na década de 60 como um campo da inteligência artificial, cujo os objetivos são aprender padrões em base de dados e criar aplicações que sejam capazes de melhorar automaticamente com a experiência (MCCARTHY, 2007). Em AM, algoritmos são desenvolvidos para descobrir relações complexas desconhecidas entre variáveis de um grande conjunto de dados e estes algoritmos produzem resultados baseados no que eles descobriram ou "aprenderam". É importante ressaltar que existem muitos algoritmos de AM que diferem um do outro no método que usam para descobrir padrões (BALOGLU *et al.*, 2021).

### **2.2.2 *Visão Computacional***

A identificação visual de objetos é trivial para humanos, mas consideravelmente desafiadora para computadores. A visão computacional como a área de estudo que tenta

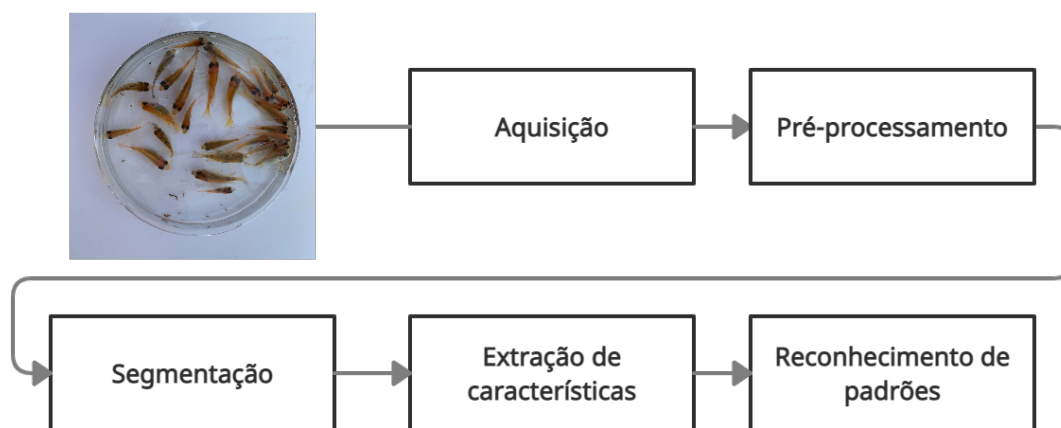
repassar para máquinas a incrível capacidade da visão, não apenas ao ato de captar imagens em si, mas um processo mais vasto e mais rico, podendo ter diversos objetivos, dentre eles: melhorar o funcionamento delas, separar regiões ou objetos de interesse de uma cena e extrair informações como por exemplo: forma, textura, brilho e ainda relacionar a imagem com outras vistas anteriormente (BACKES; JUNIOR, 2019).

De acordo com as fontes pesquisadas, podem existir variações na quantidade e suas atribuições, mas é uma prática comum dividir as fases de um sistema de visão computacional em:

- Aquisição: é o primeiro passo e caracteriza-se pela captura, armazenamento e transmissão de uma imagem. Essa etapa pode ser realizada com instrumentos de diversos tipos como, por exemplo, *scanner*, câmeras digitais, infravermelho, celulares, *smartphones*, *tablets*, câmeras de segurança normais ou térmicas, placas digitalizadoras, *webcams*, câmeras para 3D, satélites, ressonância magnética, tomografia computadorizada e outros (BORTH *et al.*, 2014);
- Pré-processamento: responsável por utilizar técnicas de pré-processamento para aprimorar a qualidade da imagem, isto é, retirar ruídos, salientar bordas, suavizar imagem podendo ainda realizar processamentos mais complexos como: filtros e operadores morfológicos (BACKES; JUNIOR, 2019);
- Segmentação: conforme Borth *et al.* (2014), esta etapa tem por finalidade separar a imagem em regiões de interesse, baseando-se no problema a ser resolvido, como a separação da região do objeto e o fundo da imagem. A depender do problema, caso esteja relacionado a contagem ou o reconhecimento de muitos objetos, pode, além de separar os elementos irrelevantes separar também os objetos de interesse em regiões distintas;
- Extração de características: é um processo responsável por obter um conjunto de características do objeto de interesse, como detecção de bordas, cantos ou pontos, através da procura de codificações numéricas (BACKES; JUNIOR, 2019);
- Reconhecimento de padrões: configura-se por meio do reconhecimento de faces, defeitos, estruturas, objetos ou qualquer alvo que possa ser classificado na imagem. Ele está diretamente relacionado a encontrar padrões ou regularidades, logo essa etapa tem por objetivo rotular ou agrupar a imagem com base em seu conjunto de características (RUDEK *et al.*, 2001).

Na Figura 2, é apresentado um esquema simples das etapas de um sistema de visão computacional.

Figura 2 – Esquema de um sistema de visão computacional



Fonte: autoria própria (2021).

### 2.2.3 Redes Neurais Artificiais

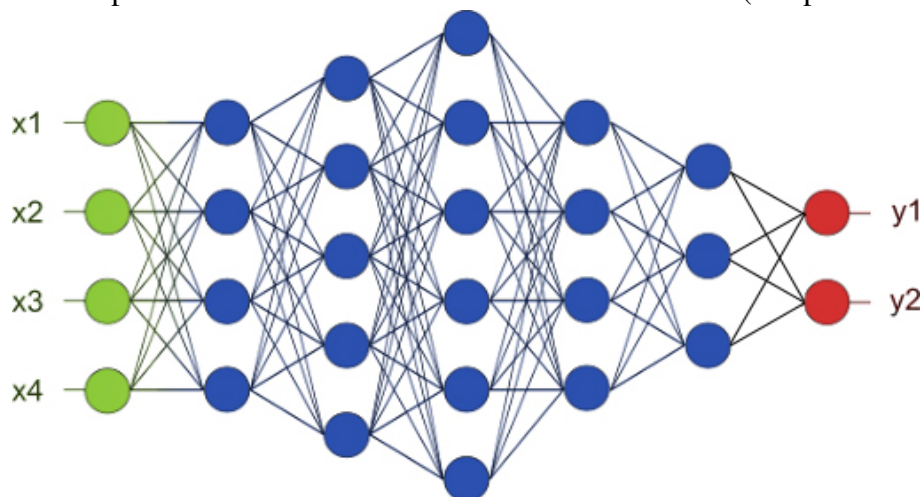
A utilização de *Redes Neurais Artificiais* (RNAs) vem sendo bastante difundida nos últimos anos, isso graças ao aumento do poder computacional e à utilização das unidades de processamento gráfico (GPU), o que possibilitou o acesso ao processamento paralelo mais barato e acessível, melhorando significativamente o desempenho. RNA têm suas arquiteturas e comportamentos inspiradas no cérebro humano, baseando-se principalmente no funcionamento e nas conexões que os neurônios realizam, sendo capaz de, a partir de um conjunto de dados reduzido, aprender através da experiência, generalizando, assim, a informação e dando respostas coerentes para dados ainda não conhecidos (KOVÁCS, 2002).

Morais (2010) define RNA como sistemas paralelos e distribuídos, compostos por unidades de processamento responsáveis por calcular determinadas funções matemáticas. As unidades são conectadas e suas conexões possuem pesos, os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida pelos neurônios da rede. As unidades são distribuídas em uma ou mais camadas interligadas por um grande número de conexões.

### 2.2.4 Aprendizagem Profundo

Aprendizagem Profunda (*deep learning*), é uma sub-área de aprendizado de máquina. Neurônios matemáticos são unidades, responsáveis por calcular a soma ponderada de várias entradas, aplicar determinada função e passar o resultado para próxima camada. Por meio de varias camadas de neurônios matemáticos, as redes de aprendizagem profunda são capazes de realizar diversas tarefas, como: processar dados, compreender a fala humana e reconhecer objetos visualmente (Data Science Academy, 2021). A informação é processada através de cada camada e a saída da camada anterior fornece entrada para a seguinte, conforme exibido na Figura 3. Essas camadas são tipicamente dividida em camada de entrada, também chamada de camada visível por conter as variáveis que se pode observar, que são os nós da primeira camada, destacados na Figura 3 em verde, sendo responsáveis por receber os dados; camadas ocultas, chamadas desse modo por seus valores não serem recebidos diretamente da origem dos dados ficando entre as camadas de entrada e saída, que são responsáveis por extrair características, destacados na Figura 3 em azul; e, por fim, a camada de saída, na qual é apresentada a saída da rede, conforme mostra-se na última posição, com a cor vermelha.

Figura 3 – Exemplo de uma Rede Neural Artificial Multicamadas (Deep Learning)



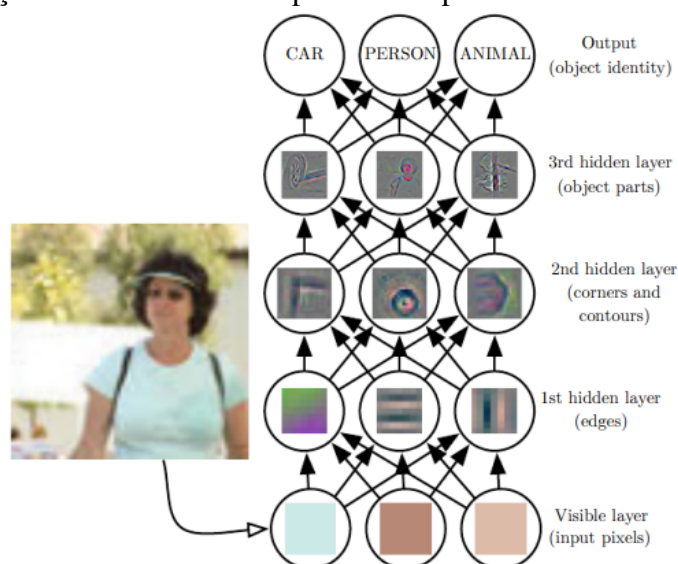
Fonte: autoria própria (2021)

Uma das dificuldades enfrentadas por algoritmos de IA é que alguns problemas do mundo real possuem muitos fatores de variação para cada trecho dos dados que devem ser analisados. O aprendizado profundo trata esse problema de representação por meio de análises que são feitas a partir de representações mais simples. Aprender a representação

certa dos dados em um nível profundo permite que o aprendizado seja realizado em várias etapas.

A Figura 4 mostra como um sistema de aprendizado profundo pode representar o conceito de uma imagem de uma pessoa combinando conceitos mais simples, como cantos e contornos que, por sua vez, são definidos em termos de bordas e, por fim, a saída é a classificação da imagem. Conforme Goodfellow *et al.* (2016), o aprendizado profundo é um tipo particular de aprendizado de máquina com grande poder e flexibilidade, já que constrói a aprendizagem por meio da representação do mundo com uma hierarquia aninhada de conceitos, sendo estes definidos em relação a conceitos mais simples e representações mais abstratas, calculados em termos de outros menos abstratos.

Figura 4 – Ilustração de um modelo de aprendizado profundo



Fonte: Extraído de (GOODFELLOW *et al.*, 2016)

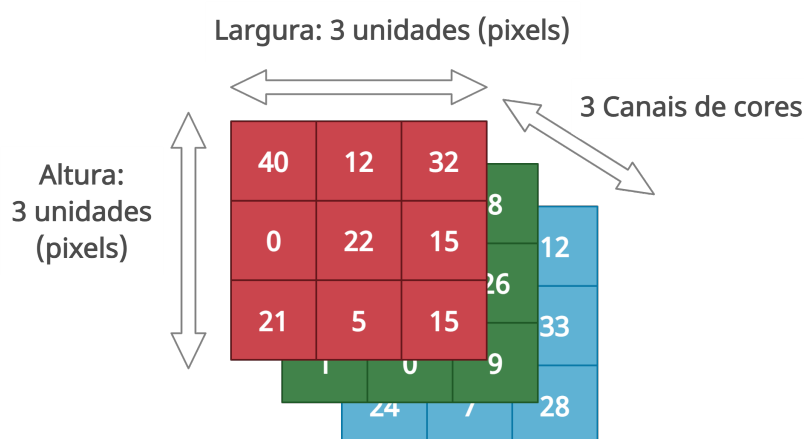
### 2.2.5 Redes Neurais Convolucionais

Data Science Academy (2021) define *Redes Neurais Convolucionais* (CNN) como um algoritmo de aprendizado profundo, que pode captar uma imagem de entrada, atribuir importância (pesos e vieses que podem ser aprendidos) a vários aspectos e ser capaz de diferenciá-los um dos outros. O pré-processamento exigido em uma CNN é muito menor em comparação com outros algoritmos de classificação, pois, nos outros os filtros são disponibilizados de forma manual, com treinamento suficiente, mas as CNN têm a capacidade de os aprender e extrair as características por si só.

Na Figura 5, é apresentado a representação de uma entrada de uma imagem RGB

(*red, green, blue*), cujos dados da imagem são três matrizes, sendo que a altura e a largura delas dependem diretamente da altura e largura da imagem. Em outras palavras, tem-se um conjunto de dados tridimensional construído pela representação de cada cor por uma matriz cujos valores normalmente são atribuídos de acordo com os *pixels*.

Figura 5 – Representação de uma entrada de imagem RGB



Fonte: autoria própria (2021)

As CNN são comumente compostas de camadas de convoluções e *pooling*, conforme Vargas *et al.* (2016), sendo a primeira composta por vários neurônios, onde cada um é responsável por aplicar filtros em pedaços específicos da imagem e a segunda por reduzir a dimensionalidade dos dados recebidos, através da diminuição da altura e largura da entrada. Tendo também camadas de *flattening*, que são encarregadas de fazer a redução de dimensões dos dados recebidos da camada anterior, mudando por exemplo uma entrada de duas dimensões para uma (SIDDIQUE *et al.*, 2019).

Atualmente, CNNs já são aplicadas aos mais diversos problemas de visão computacional, sendo utilizadas para serviços de pesquisa de imagens, contagem de objetos, carros autônomos, identificação de doenças, sistemas automáticos de classificação de vídeo etc. Pode-se, ainda, trabalhar com esses dados para reconhecimento de padrões em áudio, reconhecimento de voz e processamento de linguagem natural.

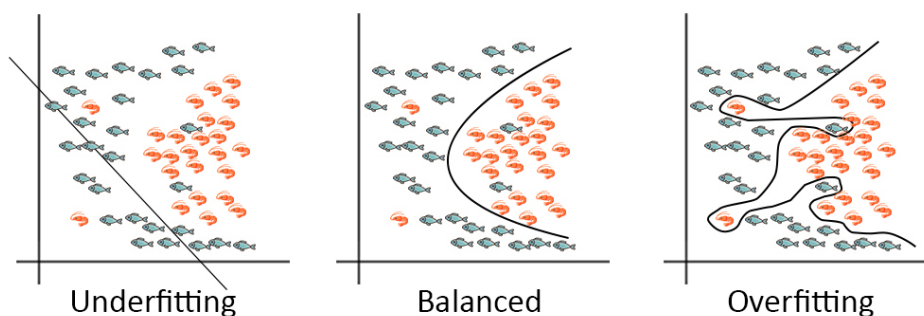
### 2.2.6 Overfitting

Quando se realiza o treinamento de redes neurais, deve-se ficar atento ao *overfitting*, um processo que acontece quando o modelo, no momento do treinamento, ajusta-se excessivamente ao conjunto de dados, tendo assim um bom desempenho na base de



treinamento, mas um desempenho ruim ao analisar novos dados (JABBAR; KHAN, 2015). Já o *underfitting* é quando o modelo já é ruim no treinamento e não consegue tornar-se satisfatório. A atenção a esses elementos é importante porque o objetivo final é a criação de um modelo balanceado, que possa aprender com a base de treinamento e generalizar bem, obtendo bons resultados quando apresentado a dados novos. Na Figura 6 são apresentados gráficos que exemplificam os três conceitos.

Figura 6 – Exemplo de Overfitting



Fonte: autoria própria (2021)

### 3 TRABALHOS RELACIONADOS

O presente trabalho desenvolveu dois tipos de modelos de visão computacional distintos, sendo o primeiro voltado para detecção de *outliers* e o segundo com o objetivo de contagem de animais na aquicultura. Em vista disso, a fim de conhecer melhor o domínio dos problemas e como foram criadas soluções pela comunidade científica, este estudo selecionou e analisou os alguns trabalhos sobre os temas referidos.

#### 3.1 Detecção de *outliers*

Conforme Hendrycks *et al.* (2018), é importante a detecção de entradas anômalas ao se trabalhar com modelos de predição, sobretudo com aprendizado profundo, pois entradas que são *outliers* ampliam a dificuldade e comprometem os resultados gerados pelos modelos. Logo, muitas vezes se faz necessária a aplicação de algoritmos que filtrem as imagens que serão utilizadas pelos modelos finais. Essa etapa tem por objetivo diminuir o tempo de processamento e aumentar a precisão do modelo final.

Análise da aplicação dessa abordagem são apresentados em trabalhos como, por exemplo o de Rahimzadeh *et al.* (2021), que propõe um sistema capaz de detectar COVID-19 com base em imagens de tomografia computadorizada do paciente. Nesse estudo foi utilizado um conjunto de dados que inclui 48260 imagens de tomografia computadorizada de 282 pessoas sem COVID-19 e 15589 imagens de 95 pacientes infectados pela COVID-19. Para avaliar se as imagens devem ou não ser utilizadas no modelo de predição, é realizado a verificação da quantidade de *pixels* escuros de determinada área da imagem, e em seguidas classificadas como pulmão aberto ou fechado. As imagens de pulmão aberto são utilizadas pelo modelo predição e as como pulmão fechado não.

Na pesquisa (PEREZ; TAH, 2020) é trabalhada a classificação de imagens de construções em três classes com defeito, sendo: mofo, mancha e deterioração da pintura e uma classe para sem defeitos. Utilizaram o modelo *VGG-16* pré-treinado para extrair as imagens atípicas do conjunto de dados e, depois, treinaram novamente o modelo *VGG-16*, gerando resultados que mostraram uma precisão da classificação melhor, ocasionando, também, a redução no número de casos classificados incorretamente.

Um outro estudo (MONKMAN *et al.*, 2019) utilizou uma metodologia de visão de máquina para avaliar uma estimativa do comprimento total (CT) de um peixe fusiforme,

o robalo europeu, a partir de imagens públicas. Usou três redes neurais convolucionais regionais (R-CNN) treinadas. O trabalho valeu-se do algoritmo floresta de isolamento para remover *outliers* com base na razão entre altura, a largura de detecção, a pontuação e o *mean bias error* (MBE). Como resultado, constatou-se que houve uma previsibilidade, com 95% de êxito na localização dos peixes.

Em uma perspectiva semelhante aos estudos citados, o trabalho ora apresentado visa abordar as etapas do desenvolvimento de modelos que possam classificar se as imagens enviadas pelo usuário estão em conformidade ou não com as especificações definidas pelo *dataset* Vivarium.

### 3.2 Contagem de animais

Os autores Salazar *et al.* (2015), registram um método para contagem de alevinos da espécie tilápia vermelha com base em um algoritmo de visão computacional desenvolvido no *LabVIEW*, que é uma linguagem de programação gráfica. Como ferramenta de captura, desenvolveu-se uma estrutura com uma rampa e iluminação, na qual foi acoplada uma câmera de baixo custo, *webcam Genius FaceCam 1010*, com alta resolução e velocidade máxima de 30fps. O *dataset* utilizado para validar o algoritmo foi a contagem de 20 lotes de peixes e a quantidade do lote variava entre 8 e 10 animais com tamanho médio de 8mm. Ao final, os pesquisadores indicaram que a taxa de erro na contagem dos alevinos foi de 8,68% e propuseram como melhorias a utilização de equipamentos e computadores com melhores especificações, pois, apesar do aumento nos custos, tais ações poderiam trazer melhoria na precisão.

O trabalho de Hernández-Ontiveros *et al.* (2018) propõe, através de um sistema utilizando *raspberry PI 2* acoplado a uma câmera *CCD 5mp* e outros periféricos, um método para contar peixes vivos em tempo real de forma automática. Os testes exposto no estudo foram realizados com duas espécies, o guppie (*Poecilia Reticulata*) e a molinésia (*Poecilia Sphenops*), sob condições de iluminação controlada e animais com tamanho variando entre de 0,5cm a 2,3 cm. Foram utilizados um total de 11.520 imagens, das quais 300 foram selecionadas aleatoriamente como conjunto de treinamento para criar o modelo de predição. As imagens restantes foram usadas para validar e testar o método proposto, bem como criar um segundo conjunto de referência. Na contagem dos peixes, foi obtida uma precisão média de até 95,57% para o algoritmo proposto e de 96,64% para o que foi

utilizado na comparação. Apesar de o método proposto ter obtido um resultado inferior, o mesmo possui um menor custo computacional. Por fim, os pesquisadores informaram que o sistema proposto pode ser melhorado usando a nova geração de dispositivos de computação e uso de uma câmera de alta resolução.

O trabalho de (LABUGUEN *et al.*, 2012), realizou uma pesquisa que tem como principal contribuição a aplicação de técnicas de visão computacional para o desenvolvimento de um sistema de contagem de alevinos de peixe-leite e realização de um experimento para estudar a movimentação desses mesmo animais e da tilápia, quando expostos a fontes de luz de cores variadas. O sistema foi testado utilizando um computador conectado a uma câmera, a Logitech Webcam C905 com captura de vídeo em resolução de 1600x1200 e posicionada a 35cm do recipiente, onde eram colocados os animais. Foram realizados 6 testes, sendo eles de 200, 418, 550, 700 e 800 e 1000 animais. Para os quatro primeiros lotes, foi obtida uma precisão média de 95%; mas para os dois restantes, foram respectivamente 75% e 68,10%. A taxa de erro maior para os últimos dois lotes é atribuída à sobreposição dos animais, contudo os pesquisadores informam ainda que esse fator pode ser mitigado com o ajuste do tanque e o tamanho da coluna d'água.

Conforme Boksuwan *et al.* (2018), na Tailândia, o processo de cultivo de camarões é dividido em três estágios: incubatório, berçário e estágio de crescimento. Os autores desenvolveram um sistema de visão computacional que realiza a contagem de camarões com o objetivo de estimar a biomassa de tanque de cultura interno. O trabalho informa que a coleta dos animais é feita por meio de um aparelho desenvolvido pelo grupo, porém não traz as especificações do produto e nem apresenta os dados numéricos a respeito da solução, apenas afirma que teve um resultado satisfatório. É importante frisar que a pesquisa apresenta uma estratégia diferenciada por realizar a contagem sem a necessidade de um agente humano que capture os animais, já que o próprio aparelho, que está posicionado sobre o tanque, aciona um mecanismo que insere um recipiente no local, faz a captura de uma mostra de água contendo os crustáceos, move a amostra para próximo da câmera e, depois, faz a devolutiva do material para o tanque.

Khantuwan e Khiripet (2012) propõem um método de contagem de números de larvas de camarão em recipientes cheios d'água usando imagem e técnicas de processamento de imagens. Para captura essas imagens, eles sugerem o uso de uma câmera digital (modelo não informado) fixa sobre o contêiner com os animais. O algoritmo foi

implementado utilizando *MATLAB* e ferramentas de processamento de imagens como remoção de ruído e aprimoramento. A validação foi realizada por meio de um *dataset* de 100 imagens, cujo os autores citam que permitiu uma alta variação do número de larvas (1000 - 2000), posição das larvas, mudanças de iluminação, mudanças na aparência e tamanho. A precisão de acertos foi de 97% em comparação com a contagem manual.

O trabalho de Kaewchote *et al.* (2018) faz o uso do algoritmo floresta aleatória para realizar predição do número de pós-larvas de camarões, isso por meio da análise de dois formatos diferentes de imagens o formato RGB (*red, green, blue*) e LBP (*Local Binary Pattern*). O experimento utilizou um *dataset* de 100 imagens, sendo que o número de animais variou entre 74 e 200. Para criação do modelo, foi utilizada validação cruzada, com 80% dos registros processados no treinamento e os outros 20% foram usados para avaliação. O formato LBP obteve, segundo os pesquisadores, a melhor acurácia, com taxa de 98,5%. O software foi desenvolvido utilizando *OpenCV* e *Numpy*, ambos provenientes de bibliotecas da linguagem *python*.

A partir dos trabalhos selecionados, foi realizado um compilado de informações e criado a Tabela 1. Nas primeiras três linhas, são apresentados trabalhos que implementaram soluções para contagem de peixes e, nas seguintes, contagem de camarões. Conforme nota-se, existe uma variação muito grande em relação à quantidade de amostras e à quantidade de animais nos *datasets* utilizados para validação das propostas. Já em questão de linguagem de programação utilizada no desenvolvimento, o *MatLab* foi o que mais se destacou, com dois projetos. Quanto ao dispositivo utilizado para processamento, três trabalhos utilizaram computadores pessoais (PC), um utilizou *notebook* e dois optaram por utilizar placas específicas. Uma alta acurácia é um dos principais objetivos em sistemas de contagem e pode-se observar que esse fator obteve um resultado elevado em todos os trabalhos.

Dentre as informações relevantes observadas no decorrer da análise dos trabalhos, pode-se destacar a importância do controle do nível da água do contêiner que está sendo observado, pois se o nível for muito elevado, irá facilitar a sobreposição dos animais, dificultado assim o processo de contagem. Quanto ao tipo de imagem analisada, o primeiro e o terceiro utilizaram vídeos e os demais trabalhos utilizaram fotos. Com exceção do quinto trabalho que não relatou detalhes sobre o procedimento e criação do *dataset*, os demais explicitaram que utilizaram mecanismos que permitiam fixar a distância da câmera para o contêiner, e com exceção do quinto e sexto todos os demais relataram uso de

fontes de luz sobre o objeto observado. Todos os trabalhos optaram por técnicas de visão computacional, porém somente o sexto trabalho associou as técnicas juntamente com AM, por meio do algoritmo *Random Forest*.

Trabalho	Nº de Amostras	Variação	Linguagem	Dispositivo	Acurácia
Salazar 2015	20	8-10	<i>LabView</i>	PC	91,32%
Hernandez 2018	11.520	100 - 350	<i>GNU Octave</i>	<i>Raspberry</i>	95,57%
Labuguen 2012	6	200 - 1000	<i>Matlab</i>	<i>Notebook</i>	95%
Boksuwan 2018	n/a	n/a	n/a	Embarcado	n/a
Khantuwan 2012	545	1000 - 2000	<i>MatLab</i>	PC	97%
Kaewchote 2018	100	74 - 200	<i>Phython</i>	PC	98,50%

Tabela 1 – Compilado - Trabalhos sobre automação de contagem.

Os trabalhos analisados apresentam diversas abordagens para realizar a contagem de alevinos e pós-larvas, porém nenhum deles criou uma versão que se integre com sistemas web ou mecanismos que permitam a captura e processamento de imagens por meio de *smartphones*. A ausência desse recurso faz com que as soluções propostas sejam concretizadas por meio da execução de uma atividade em duas etapas, já que a foto capturada precisa ser processada posteriormente em um computador ou há uma etapa para a aquisição e a montagem de equipamentos específicos que realizem essa contagem de maneira automática.

Logo, o presente trabalho se diferencia dos aqui expostos por gerar modelos que serão inseridos futuramente em uma plataforma *web*, permitindo que os produtores capturem a imagem diretamente dos seus *smartphones* e o sistema faça a contagem, sem a necessidade que os mesmos tenham que alimentar o modelo em computadores ou tenha que adquirir equipamentos especializados. Além disso, é importante pontuar a criação de uma metodologia que permite a inclusão de novos subconjuntos com espécies de animais distintos e em tamanhos diferentes. Quanto à contribuição na área de ciência da computação, destaca-se o desenvolvimento e avaliação de novos modelos de predição utilizando *deep learning*. Em relação aos resultados, este trabalho mostrou uma acurácia superior obtendo 98,66% conforme o material apresentado na seção 6.2.

## 4 METODOLOGIA

Detalha-se, aqui, os procedimentos metodológicos que foram utilizados para se atingir os objetivos propostos na seção 1.3, sendo organizados da seguinte forma: revisão de literatura, análise da pesquisa, criação de *datasets* e desenvolvimento da solução. A fim de se obter um melhor resultado, todas as etapas estão sendo realizadas utilizando a metodologia SCRUM.

- Revisão de literatura - consistiu na seleção de referencial teórico para fornecer embasamento para execução do projeto, tendo três principais linhas de pesquisa: mercado de aquicultura, buscando por trabalhos que relatassem a realidade do setor a partir de uma perspectiva *top-down*, que é uma abordagem que busca inicialmente por trabalhos que abordem o tema de modo geral e em seguida afunila para assuntos de níveis específicos; áreas da computação, que corroboram a produção do trabalho e técnicas de visão computacional; e aprendizagem de máquina aplicadas à contagem de alevinos e pós-larvas;
- Análise da pesquisa - etapa responsável por apresentar os resultados de explorar, refletir, comparar e gerar conhecimento a partir do material coletado na etapa anterior. Por meio da análise do material, foi possível aprender acerca das tecnologias utilizadas, compreender o mercado a qual o produto está destinado, identificar os principais desafios enfrentados e, por fim, mapear os requisitos necessários para o desenvolvimento dos *datasets* e da solução;
- Criação de *datasets* - momento em que se registra uma metodologia para criação dos *datasets*, compostos por vários subconjuntos, ampliando assim as possibilidades dos modelos que foram treinados a partir dele;
- Desenvolvimento e avaliação da solução - desenvolvida em três partes, a saber, a criação e validação dos modelos de predição, desenvolvimento da API MIC para que os modelos possam ser acessados e, por fim, a alteração na plataforma Crevettic, para que a mesma possa se comunicar com o MIC;
- Avaliação dos modelos - os modelos para detecção de *outliers* foram avaliados pelas métricas *accuracy*, *precision* e *recall*, enquanto os modelos de contagem com *accuracy*, *Mean Average Precision* (mAP) e *Root Mean Squared Error* (RMSE).

## 4.1 SCRUM

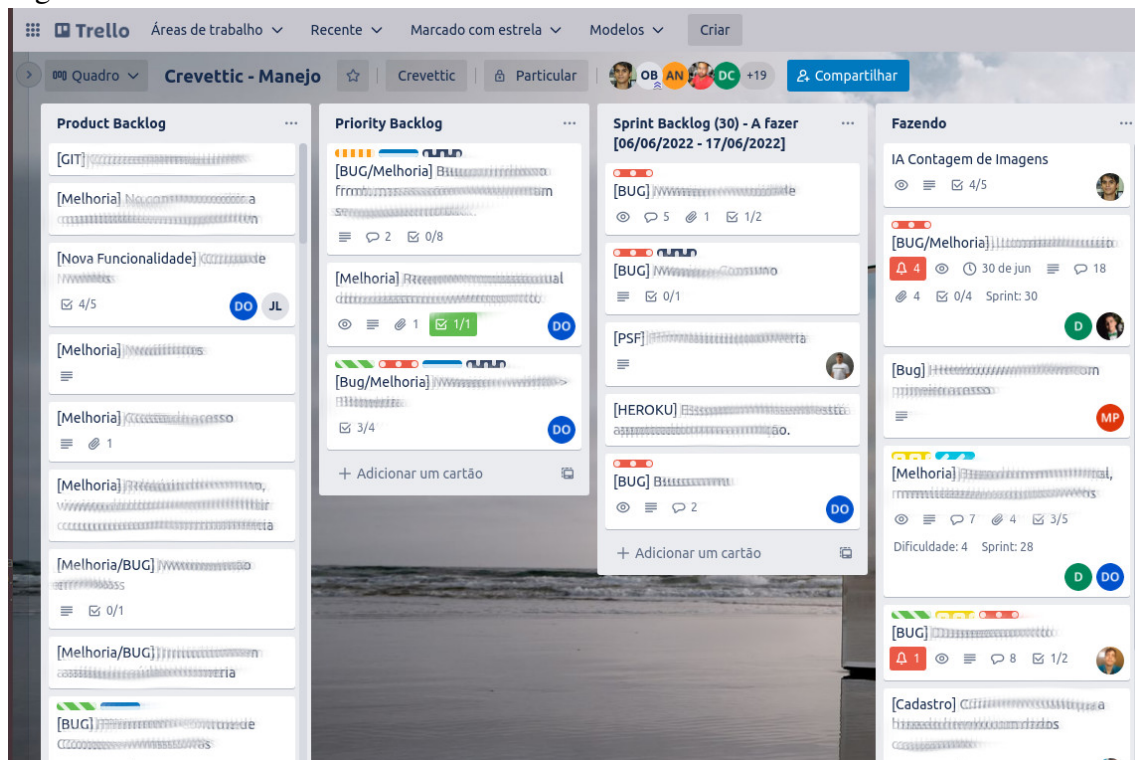
Para gestão do projeto está sendo utilizado a metodologia SCRUM. Sutherland *et al.* (2007) a define como um procedimento ágil de desenvolvimento de *software* projetado para adicionar energia, foco, clareza e transparência para equipes que atuam nesse processo. Apesar de ser projetado inicialmente como um método para desenvolvimento de software, o mesmo vem sendo cada vez mais utilizado para gerenciar projetos dos mais variados tipos e isso se deve pelo fato de que, cada vez mais, se buscam as vantagens de se usar métodos ágeis, como: aumento de previsibilidade, diminuição de riscos, além da transparência e principalmente adaptações ágeis diante de mudanças.

O método SCRUM pode ser adaptado de acordo com os interesses e os objetivos de cada organização. Porém, algumas coisas são práticas comuns do método, como a definição do escopo do projeto à medida que são realizadas as entregas, já que cada uma delas é desenvolvida através de ciclos iterativos conhecidos como *Sprints*. Um *sprint* é um período de tempo limitado que, ao seu final deve-se entregar uma nova versão ou funcionalidade. Isso garante que, mesmo nas primeiras *sprints*, já se tenha um produto utilizável, garantindo um *feedback* dos usuários para, assim, realizar ajustes no escopo e um refinamento do produto.

Vale ressaltar que todas as atividades são trabalhadas em conjunto com a equipe Crevettic e com parceiros do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), campus Aracati. Logo, o projeto contou com a constante validação e instruções de especialistas da área de ciência da computação e de consultores da área de carcinicultura e piscicultura. Seguindo a metodologia de gestão de projetos da Crevettic, as atividades de desenvolvimento foram geridas utilizando o *Trello*, como ferramenta de apoio, conforme a Figura 7.



Figura 7 – Ferramenta Trello



Fonte: autoria própria (2021).

## 5 SOLUÇÕES

Com base na revisão de literatura e nos levantamentos de requisitos com especialistas de aquicultura e fazendeiros, foi possível entender o funcionamento do processo de contagem manual de animais e os principais desafios da automação dessa atividade. Logo, para se alcançar os objetivos propostos, primeiramente foi preciso definir como seria o processo de captação e de organização das imagens. Posteriormente foram criados modelos de predição para avaliar se as imagens seguem as especificações definidas e em seguida foram desenvolvidos modelos de contagem.

### 5.1 *Design Vivarium*

A criação de modelos de predição eficientes está diretamente ligada à qualidade e quantidade dos dados utilizados durante o processo de treinamento e validação. Ao realizar a revisão literária, não foi possível encontrar um conjunto de dados públicos que permitisse a criação dos modelos propostos. Logo, para atender a essa demanda e às futuras, foi desenvolvida a primeira versão do conjunto de dados Vivarium, o mesmo tem por objetivo reunir fotos de animais enquanto estão em processo de contagem, para assim permitir o aprendizado e a validação de modelos de aprendizado de máquina.

Como os modelos propostos tem por objetivo serem utilizados sem a necessidade de nenhuma estrutura de fixação complexa ou simplesmente sem nem um apoio para câmera ou *smartphone*, urge uma série de procedimentos mínimos para que as fotos mantenham uma padronização, conforme será abordado na seção 5.1.2. Durante a fase aquisição de imagens para o *dataset*, as que não seguem as especificações podem ser facilmente descartadas, porém, quando o sistema de contagem estiver em produção, pode ser que o usuário envie fotos inválidas. Logo, para solucionar essa limitação, será empregado um modelo que verifica se a imagem está ou não em conformidade e, por conta disso, além das fotos em conformidade cada subconjunto pode conter também fotos em não conformidade.

A atividade de contar animais pode ser realizada em várias fases ao longo do processo de cultivo, portanto, o tamanho dos animais pode variar a depender da etapa e das características da espécie observada. Logo a captura de imagens de uma única variação de tamanho não garantiria um bom aprendizado, além de impossibilitar a validação dos resultados para diferentes produtores, tendo em vista que em uma mesma fazenda pode-se

realizar a contagem em diferentes estágios de cultivo. Dessa forma, a criação de modelos que possam lidar com essa grande quantidade de variações se faz necessária e, por isso, o Vivarium deve ser construído a partir de um compilado de subconjuntos que variam espécie, tamanho, câmara, local etc, sendo que todos os subconjuntos são construídos seguindo as especificações a seguir.

### 5.1.1 Organização

Com objetivo de criar um *dataset* escalável e de fácil manipulação, foi definido um conjunto de regras a serem seguidas no processo de inclusão de novos subconjuntos. O *dataset* será disponibilizado na seguinte arquitetura: uma pasta nomeada de *Vivarium* e, dentro dela, deve-se disponibilizar as planilhas *info\_vivarium\_dataset*, *info\_vivarium\_files* e as pastas onde estão armazenados os subconjuntos. A nomeação do subconjunto é dada pelo nome da espécie seguido da quantidade atual de subconjuntos da espécie.

id	nome	range	Anotação	Conformidade		Recipiente	Câmera
				sim	não		
1	tilapia_001	5-30	111	111	111	13cm	Iphone Xs
2	tetra_001	1-9	63	63	117	20cm	Pocophone F1
3	tilapia_002	0-60	183	183	183	13cm	Galaxy Note 9
4	tambacu_001	0-30	95	95	148	13cm	Pocophone F1
5	tilapia_003	0-152	685	1916	152	25cm	Note 9 e Moto E7
6	tilapia_004	50-60	63	63	9	26cm	Note 9 e Moto E5
7	tilapia_005	51-60	20	20	0	26cm	Samsung A51
8	vazio_001	0	589	0	589	26cm	Note 9 Samsung A32

Tabela 2 – Dados da planilha *info\_vivarium\_dataset*.

Todo subconjunto deve ser cadastrado na planilha *info\_vivarium\_dataset*, que é uma forma de gerenciar e de facilitar a manipulação dos subconjuntos. Na Tabela 2, o valor da coluna recipiente representa o diâmetro/lado do objeto utilizado para contar os animais. Ressalta-se que, além dos dados apresentados, também devem ser informados o recipiente utilizado, o tamanho médio dos animais, o local de captura, a data, os dispositivos de captura de imagens, a cidade, e o estado. A medida que novos subconjuntos foram sendo adicionados, foram realizados experimentos com diferentes tipos e tamanhos de recipientes, até o momento foram utilizados: placas Petri e fundos de recipientes plásticos utilizados para condicionamento de alimentos. Os recipientes plásticos se mostram como uma opção mais viável, pois são maiores, permitindo uma melhor distribuição dos peixes além de que

ambientes com muita iluminação não causam reflexo. Salienta-se que existe, no arquivo, também o campo observações, que deve ser utilizado apenas quando necessário.

Dentre as informações relevantes que foram inseridas em observações, pode-se citar por exemplo que no subconjunto *tilapia\_004* as fotos do aparelho *Moto E5* não estão na proporção 1x1, em razão da limitação do aplicativo da câmera. Enquanto que no subconjunto *vazio\_001*, é informado que se trata de um subconjunto em que não são fotografados animais, apenas o recipiente buscando variar a luminosidade e o enquadramento.

Toda imagem em conformidade deve ser cadastrada na tabela *info\_vivarium\_files* que tem os seguintes campos: identificador, identificador do subconjunto, anotação, responsável pela anotação e quantidade. Dentro da pasta de cada subconjunto, as imagens devem ser dispostas de acordo com sua classificação, sendo 1 para conformidade e 0 para não conformidade.

O nome do arquivo segue o padrão proposto na Figura 8, cuja primeira parte deve ser inserida uma abreviação do nome do *dataset*, na segunda, o identificador do subconjunto; na terceira, a classe a que pertence, e por último, o identificador do arquivo.

Por ser um *dataset* escalável com sua construção sendo desenvolvida de forma descentralizada, foi desenvolvido, também, um conjunto de funções em *python* que permite verificar se os dados informados nas planilhas estão de acordo com os arquivos disponibilizados. Dentre as verificações realizadas e já implementadas, pode-se citar a checagem da quantidade de arquivos, padronização dos nomes, comparação da quantidade de animais informados na planilha *info\_vivarium\_files* e registro da quantidade identificada nas anotações.

Figura 8 – Padrão utilizado para nomear as imagens.

$$\underbrace{\text{vd}}_1 \underbrace{\text{XXX}}_2 \underbrace{\text{Y}}_3 \underbrace{\text{ZZZZZZZ}}_4 .\text{jpg}$$

Fonte: Autoria Própria (2021)

### 5.1.2 Especificações de captura de imagens

Um fator importante relatado durante a criação de *dataseset* que objetiva a contagem de peixes e pós-larvas é a padronização do processo de captura da imagem, conforme os trabalhos (SALAZAR *et al.*, 2015; LABUGUEN *et al.*, 2012; HERNÁNDEZ-ONTIVEROS

*et al.*, 2018; KHANTUWAN; KHIRIPET, 2012; KAEWCHOTE *et al.*, 2018). Dentre as dificuldades encontradas, pode-se destacar a necessidade de um ambiente controlado para que se possa fazer a aquisição das imagens, pois fatores como, por exemplo, distância, movimento, iluminação e profundidade da coluna d'água, refletem diretamente na qualidade das predições. Logo, a fim de mitigar parte dessas dificuldades, foi desenvolvido um método para auxiliar o processo de captura de imagens, sem a necessidade de grandes investimentos. Houve também criação de uma padronização para construção dos subconjuntos, pois os mesmos estão sendo desenvolvidos em ambientes heterogêneos e por equipes descentralizadas.

Ao se capturar uma imagem no processo de contagem, se ela ficasse muito próxima do alvo, era provável que se conseguisse visualizar o recipiente em sua totalidade, o que poderia deixar parte dos animais de fora da contagem, tendo em vista que parte do recipiente poderia não estar contido na foto. Contudo, se ela se afastar muito, a imagem passa a conter outros elementos e, com o aumento da distância, a visualização dos animais ficaria prejudicada. Para garantir um enquadramento adequado, sem a obrigatoriedade de um suporte fixo, foi proposta a solução exposta a seguir.

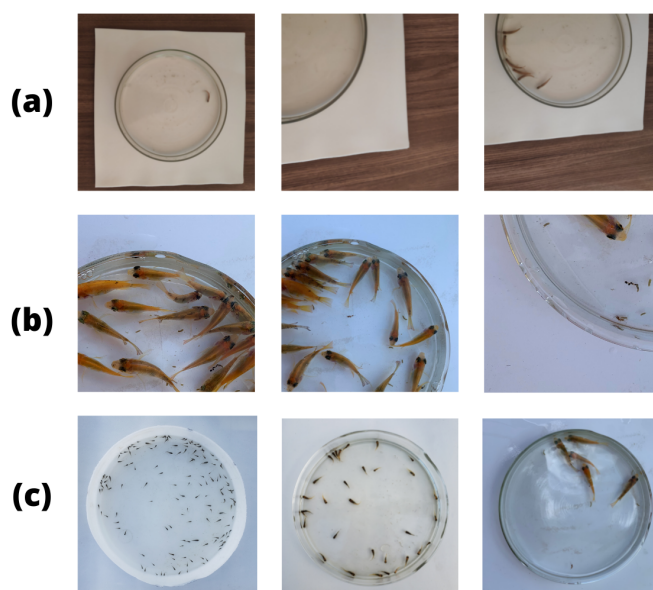
Dado o tamanho do recipiente de captura, deve-se criar uma base branca quadrada e os lados devem ser definidos por:  $L = X + (X/2)$ , sendo  $X$  o diâmetro, caso o recipiente seja circular, ou a medida dos lados, caso tenha formato de quadrado. Uma sugestão de material é a utilização de papel de etileno acetato de vinila (EVA), pois não é danificado pelo contato com a água. Caso o local de colocação da base branca também seja branco, deve-se colocar uma outra base com cor diferente, por baixo da base, com o objetivo de criar um contraste. Quanto ao recipiente, o mesmo pode ser um contêiner translúcido como, por exemplo, a placa de Petri ou uma vasilha de plástico ou, até mesmo, um aquário quadrado, desde que atente à dispersão dos animais para que não exista sobreposição. A relação entre a quantidade de animais e o tamanho do recipiente é um fator fundamental na dispersão, evitando, assim, a sobreposição, conforme os experimentos realizados.

A câmera deve ser configurada para a proporção 1:1, com objetivo de se obter imagens quadradas e facilitar, assim, o ajuste do enquadramento. Deve-se buscar, então, um ambiente com iluminação adequada, pois, se for demasiadamente escuro, haverá prejuízo na visualização; e, se for muito claro, pode haver o efeito de estouro de claridade na foto. Esses requisitos visam facilitar o processo de enquadramento ideal por parte da equipe

responsável, além de servir como parâmetros para classificar a imagem em conformidade ou em não conformidade, ressaltando-se que uma imagem é considerada em conformidade quando o recipiente é apresentado por inteiro e não possui bordas escuras na foto; e em não conformidade quando a placa não aparece por inteiro ou aparece bordas escuras na foto.

A Figura 9 exemplifica bem situações de não conformidade e conformidade. O fato do recipiente aparecer todo na imagem, não garante uma imagem de conformidade, pois ela pode estar falhando em relação à distância, como na primeira imagem da fileira (a). Ainda na primeira fileira, pode-se ver uma imagem em que aparece o fundo, sem ser o do EVA branco, e que não apresentem a placa inteira, logo essas imagens não estão em conformidade. Já a fileira (b), exibe imagens que estão perto demais do objeto, o que inviabiliza o processo de contagem, pois não se pode mensurar o que está fora da imagem. A última fileira apresenta amostras em conformidade, pois são imagens em que aparece todo o recipiente e não aparece nenhuma borda escura.

Figura 9 – Exemplos de imagens em não conformidade e conformidade.



Fonte: Autoria Própria (2021)

Caso, mesmo assim, o usuário opte por usar uma estrutura para fixar o celular, o mesmo também deve seguir as recomendações anteriores. Durante os procedimentos de captura, testou-se o uso de um modelo disponibilizado pelo IFCE e apresentado na Figura 10, que é um apoio construído em impressora 3D. O mesmo se mostrou de fácil utilização, podendo facilmente ser acoplado a diversos modelos de celulares e mesas. Na

Figura 10 também é exibida a estrutura empregada durante os procedimentos de criação de subconjuntos: um recipiente plástico, a base de EVA branca e uma base preta para fazer o contraste.

Figura 10 – Apresentação do apoio construído em impressora 3D.



Fonte: Autoria Própria (2021)

### 5.1.3 Subconjuntos

A primeira versão do Vivarium é composta por oito subconjuntos: tilápia\_001, tetra\_001, tilápia\_002, tambacu\_001, tilapia\_003 e tilalia\_004, tilapia\_005 e vazio\_001. Dessa forma, são os primeiros subconjuntos das espécies tilápia, tetra e tambacu. O tilápia\_001, tilápia\_002, tilápia\_003, tilápia\_004 e tilapia\_005 foram criados através de uma parceria com o IFCE; já o tetra\_001 e o tambacu\_001, foram criados a partir de animais de uma fazenda localizada no Acre. Mais dados a respeito das bases podem ser visualizados na Tabela 2.

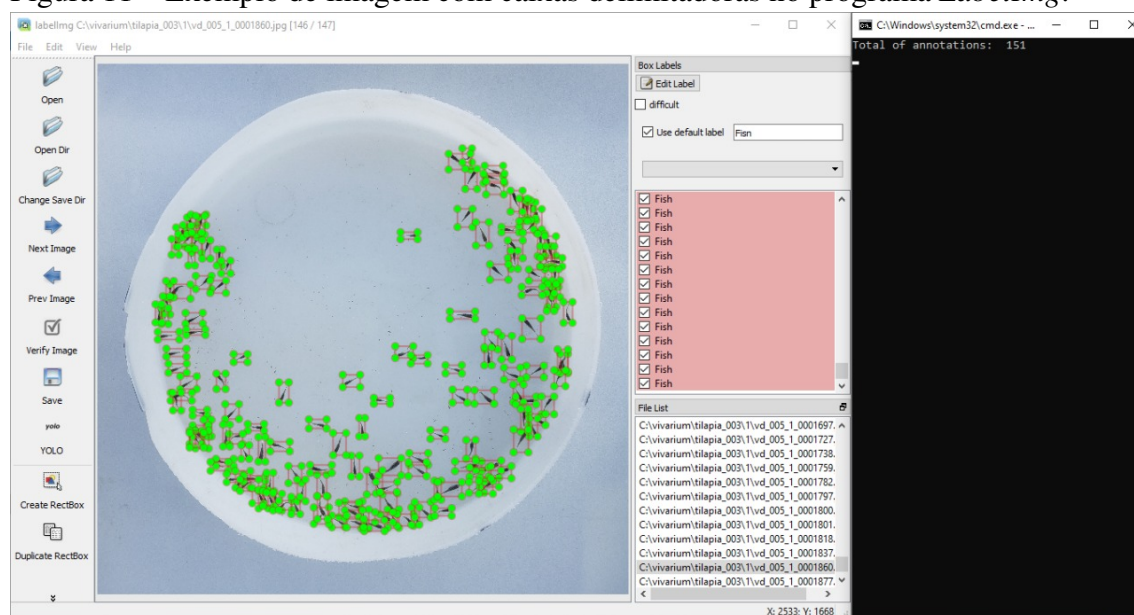
### 5.1.4 Pré-processamento

Para realizar o treinamento dos modelos baseados em rede *Yolo*, há a necessidade de realizar um pré-processamento das imagens, que é marcar caixas delimitadoras em torno de cada item que se deseja identificar. Para cumprimento dessa atividade, foi realizado um *fork* no repositório *LabelImg* (TZUTALIN, 2022) e o código foi alterado para que, a cada inserção de caixa delimitadora, fosse exibida a quantidade de caixas inseridas até

o momento no terminal. Essa alteração visa facilitar o processo de rotulação, tendo em vista que o responsável pode facilmente acompanhar a quantidade de animais inseridos à medida que realiza o processo.

Na Figura 11, expõe-se um exemplo de imagem do Vivarium com as caixas delimitadoras dentro do programa *Labellmg* e com a exibição da quantidade de caixas inseridas no terminal.

Figura 11 – Exemplo de imagem com caixas delimitadoras no programa *Labellmg*.



Fonte: Autoria Própria (2021)

Em função das limitações de tempo e de recursos humanos, não foi realizado o pré-processamento de todas as imagens em conformidade. Ao total, fez-se a rotulagem de 1220 imagens, restando ainda 1231. Na Figura 12, é mostrada a distribuição das imagens pré-processadas e as que estão aguardando processamento, todas elas agrupadas de acordo com a quantidade de animais.



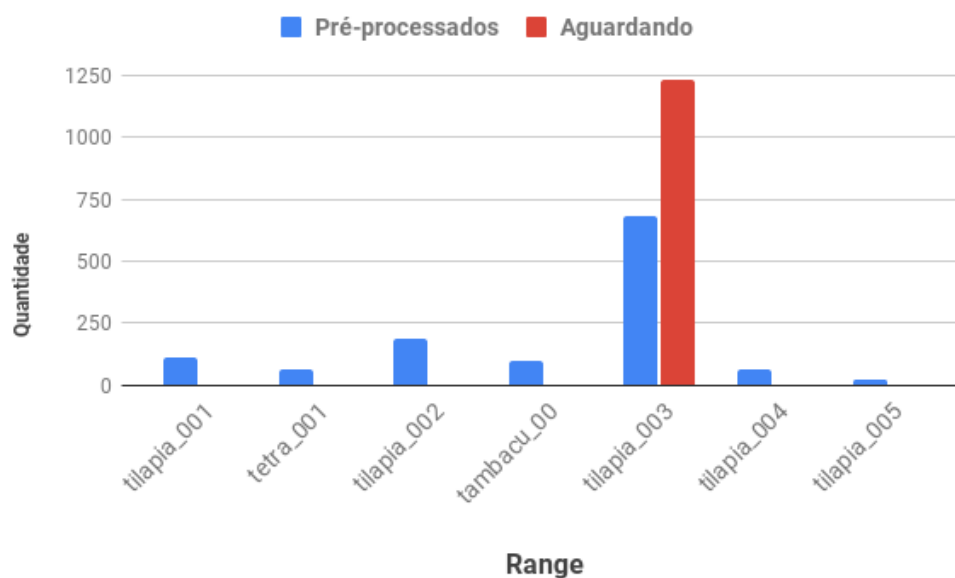
Figura 12 – Gráfico de imagens pré-processadas e as que estão aguardando processamento por quantidade de animais.



Fonte: Autoria Própria (2021)

Na Figura 13, exibe-se o gráfico com a quantidade de imagens pré-processadas por subconjuntos, respectivamente, os subconjuntos tilapia\_001, tetra\_001, tetra\_002, tambacu\_001, tilapia\_003, tilapia\_004 e tilapia\_005 possuem 116, 63, 183, 95, 685, 63 e 20 imagens pré-processadas. Somente o subconjunto tilapia\_003 possui imagens aguardando pré-processamento, sendo a quantidade de 1231.

Figura 13 – Gráfico de imagens pré-processadas e aguardando por subconjuntos.



Fonte: Autoria Própria (2021)

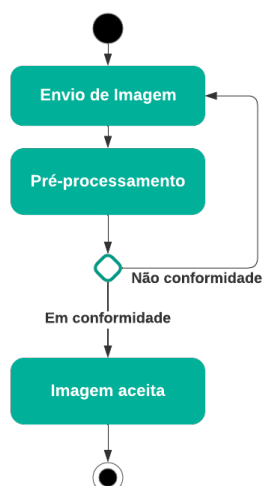
## 5.2 Modelo de Avaliação de Imagens Vivarium - MAIV

No momento de captura de imagens para a criação do *dataset*, caso o enquadramento não esteja seguindo o definido na seção 5.1.2, as imagens podem ser facilmente descartadas por meio de uma análise visual por parte dos responsáveis pelo experimento. Porém, quando o sistema de contagem estiver disponível para o usuário, apesar de serem informadas todas as instruções, pode ser que o usuário envie imagens inválidas ou com o enquadramento errado. Uma imagem fora de padrão pode ser considerada um *outliers* e fazer com que o sistema de contagem tenha o resultado de predição comprometido.

Então para garantir que o modelo de contagem só receba fotos de acordo com as especificações definidas na seção 5.1.2, foi desenvolvido um conjunto de modelos de predição que serão utilizados como filtro do que será enviado ou não para o modelo de contagem.

Quando o sistema de contagem estiver em uso, pretende-se utilizar o modelo de avaliação com melhor desempenho para filtrar as imagens enviadas pelo usuário antes de submeter ao modelo de predição de contagem. O modelo desenvolvido será utilizado como um limiar de ativação ( *threshold*), conforme diagrama de atividade exposto na Figura 14. Logo, quando o usuário enviar uma imagem, ela será pré-processada e, em seguida, será avaliada pelo modelo MAIV que, ao encontrar a imagem em conformidade, aceitará e fará o envio da mesma para ser avaliada pelo modelo de contagem. Caso contrário, o sistema solicitará uma nova amostra para o usuário. Essa etapa visa garantir uma padronização maior das imagens a serem enviadas para os modelos de contagem, gerando, assim, resultados mais precisos e eficientes.

Figura 14 – Diagrama de atividade para análise de conformidade.

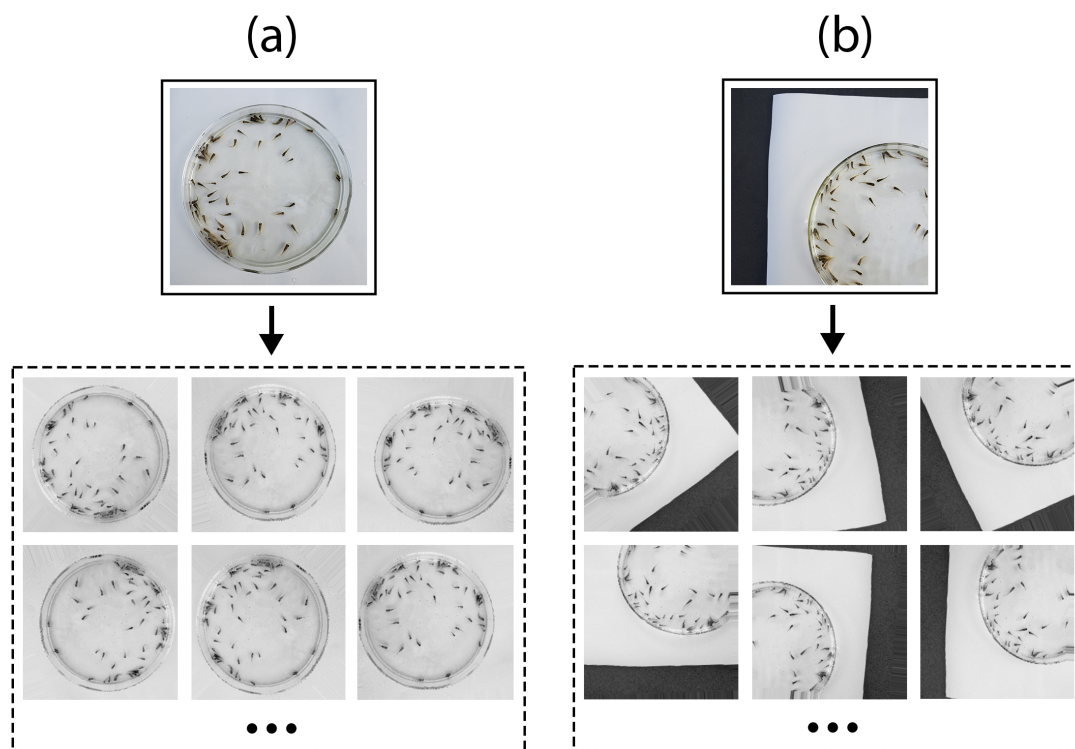


Fonte: autoria própria (2021)

Os modelos do MAIV foram desenvolvidos utilizando os subconjuntos tilapia\_001, tetra\_001, tilapia\_002 e tambacu\_001 do *dataset* Vivarium, contando com 559 imagens classificadas como em não conformidade e 452 em conformidade, totalizando 1011 imagens. Para garantir um melhor treinamento e validação do modelo, a base foi dividida em três subconjuntos, sendo eles 60% para treinamento, 20% para validação e 20% para teste, estratégia empregada também pelos trabalhos (RAN *et al.*, 2019; WONG *et al.*, 2021). Os modelos foram trabalhados e validados utilizando imagens em escala de cinza, pois assim diminui o tempo necessário para treinamento e utilização.

Conforme (VOGADO *et al.*, 2019), redes neurais convolucionais podem possuir milhões de parâmetros e precisam de uma grande quantidade de imagens para serem treinadas, dessa forma, o aumento do conjunto de dados é uma técnica que permite a criação de novos conjuntos a partir de um já existente. A API Keras fornece uma grande quantidade de parâmetros que podem ser personalizados para geração de novas imagens, como, por exemplo, rotação, inversão horizontal e vertical, movimento na horizontal e vertical e *zoom*. Na Figura 15, expõem-se exemplos da aplicação das técnicas de aumento de dados, utilizando a API Keras, sendo que, na coluna (a) são geradas novas imagens a partir de uma imagem em conformidade e na (b) são imagens provenientes de uma em não conformidade.

Figura 15 – Exemplos de aplicação de aumento de dados usando a API Keras.



Fonte: autoria própria (2021)

A fim de obter um modelo com melhor desempenho e realizar uma avaliação das redes implementadas, optou-se pelo uso de três modelos de redes convolucionais. Os parâmetros utilizados na configuração das redes e na geração de novas imagens foram baseados nos estudos Rahimzadeh *et al.* (2021) e Bessa *et al.* (2021), e estão expostos na Tabela 3.

Parâmetro	Valor
Batch Size	14
Optimizer	adam
Loss Function	Categorical Crossentropy
Epochs	20
Horizontal/Vertical flipping	Yes
Zoom Range	0.05
Rotation Range	360
Width/Height Shifting	0.05

Tabela 3 – Parâmetros utilizados na configuração das redes e na geração de novas imagens

A primeira rede convolucional utilizada é baseada na rede *TwoRConv* (BESSA *et al.*, 2021) que, por sua vez, tem sua arquitetura seguindo os padrões definidos por LeCun *et al.* (1998), que intercala camadas de convolução, *pooling* e *flattening* e, em seguida, é

conectada com uma rede neural multicamadas, que recebe saída da camada de *flattening* e realiza a classificação. A versão utilizada da *TwoRConv* tem três camadas de convolução e três camadas de *Max-Pooling*, sendo que, na última camada, há dois neurônios que utilizam função de ativação sigmóide.

A segunda rede utilizou a arquitetura Xception apresentada no trabalho Chollet (2017), e a terceira a Resnet50V2 (HE *et al.*, 2016). Para treinamento dessas redes, usou-se a aprendizagem por transferência de conhecimento, através do carregamento dos pesos pré-treinados das redes para o *dataset* ImageNet (DENG *et al.*, 2009). O ImageNet é composto por imagens com três canais de cores e como os modelos são treinados com imagens com escala de cinza, foi adicionada uma camada de entrada para permitir a entrada das imagens. Já a camada de saída de ambos os modelos foi substituída por uma camada de *pooling* global, responsável por converter as saídas da camada anterior em um vetor e repassar esses valores para uma rede neural densa. Ambas as redes foram conectadas a uma camada de 1024 neurônios e, em seguida, há uma camada de saída com dois neurônios com ativação *sigmoid*.

Os resultados obtidos através dos modelos são apresentados no capítulo 6.

### 5.3 Modelos de contagem

Para a criação da solução de contagem, optou-se por uma implementação de modelos baseados na rede *You Only Look Once* (YOLO), que é uma CNN capaz de prever, em tempo real e simultaneamente, várias caixas delimitadoras e probabilidade de classes para as mesmas, unificando, assim, vários benefícios sobre métodos tradicionais de detecção, sendo inclusive capaz de processar vídeos em tempo real (REDMON *et al.*, 2016). Além de que, essa arquitetura e suas variações veem alcançando excelentes resultados, apresentando um melhor desempenho para detecção na base *MS COCO dataset* que outros modelos (WANG *et al.*, 2021; BOCHKOVSKIY *et al.*, 2020; REDMON *et al.*, 2016).

Na pesquisa ora apresentada, optou-se pela utilização das versões *YOLOv4* e *Yolov4-tiny*, por serem mais consolidadas, documentadas e amplamente reconhecidas pela comunidade, sendo aplicadas em projetos em diversas áreas, como, por exemplo, na detecção e na contagem de pessoas (YU; ZHANG, 2021; DEGADWALA *et al.*, 2021; VALENIA *et al.*, 2021); na agricultura, para identificar flores e frutas (PARICO; AHAMED,

2021; YANG *et al.*, 2021; WU *et al.*, 2020); e, na aquicultura, para detecção de animais, identificação de doenças e estimativa de biomassa (ZHAO *et al.*, 2022; KUSWANTORI *et al.*, 2022; ABINAYA *et al.*, 2022).

O uso da versão *YOLOv4* teve como objetivo encontrar o equilíbrio entre a resolução da rede de entrada e o número de camadas convolucionais, de modo que se mantenha a precisão e a velocidade. Optou-se, também, pela arquitetura do *backbone CSP-Darknet53*, que é uma rede neural desenvolvida em C, em conjunto com *Spatial Pyramid Pooling* (SPP) para otimizar o campo receptivo; *Path Aggregation Network* (PAN) como método de agregação de parâmetros; e o *YOLOv3* como *head*, ou seja responsável pela saída (BOCHKOVSKIY *et al.*, 2020).

A *Yolov4-tiny* faz parte de uma série de versões YOLO, sendo voltada para dispositivos de baixo poder computacional, tendo sua arquitetura alterada para que ela possa rodar, com bom desempenho, em dispositivos com esse perfil. Essas escolhas fazem com que a *Yolov4-tiny* seja diferenciada da rede *YOLOv4*, pois utiliza como *backbone* uma versão reduzida de *CSP-Darknet53* e troca o *SPP* e *PAN* por uma *Feature Pyramid Networks* (FPN), além de outras alterações para reduzir o custo computacional (JIANG *et al.*, 2020).

Para criação dos modelos, selecionou-se os modelos disponíveis no repositório indicado em (BOCHKOVSKIY *et al.*, 2020), conforme indicação da documentação é possível mudar a ativação da opção *random*, o *max* com valor 200, modificando o valor do parâmetro *classes* das camadas *yolo* para 1 e do *filter* das camadas convolucionais anteriores à camada *yolo* para 18. Devido às limitações da *GPU* utilizada, a *subdivision* do *batch* foi alterada para 64, que é a quantidade de imagens processadas por vez durante as interações do treinamento e a resolução das camadas de entrada foi para 416 x 416, com o objetivo de evitar o erro de estouro de memória. A documentação recomenda que o modelo seja treinado por no mínimo 6000 épocas, que é a quantidade de interações em que os dados de treinamento são apresentados para a rede, ou até que o mAP estabilize ou, ainda, que a função de perda deixe de decair. Assim, os modelos foram treinados em 10.000 épocas e, após a análise dos resultados, escolhe-se o modelo com o melhor desempenho. O treinamento foi realizado em um computador com uma GPU GTX 1060 6Gb.

Os resultados obtidos através dos modelos são apresentados no capítulo 6.

## 5.4 Disponibilização dos Modelos

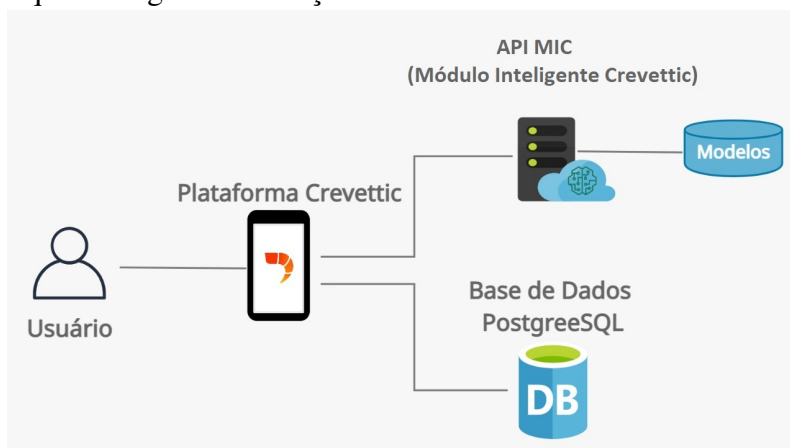
Para disponibilizar os modelos sem a necessidade de aquisição de equipamentos específicos por parte do produtor, foi implementada uma nova funcionalidade na plataforma Crevettic. Ela é acessível via computador e *smartphone* por meio de navegadores *web*. Braga (2020) define a Crevettic como um conjunto de soluções focadas na gestão do conhecimento e da inteligência de negócio de fazendas de aquicultura.

Atualmente, a ferramenta já possui diversos módulos como, por exemplo, o de gestão financeira, de gestão dos ciclos de produção e de gestão dos parâmetros da qualidade da água. Além disso, a plataforma conta com um plano de acesso gratuito, com a limitação da quantidade de funcionalidades que o usuário tem acesso, um plano *trial*, em que o usuário tem acesso a todo o sistema, mas por um tempo limitado; e um plano pago, que disponibiliza todas as funcionalidades durante o período contratado. Pretende-se disponibilizar para todos os planos, inicialmente, o acesso aos modelos desenvolvidos como produto do trabalho ora apresentado.

### 5.4.1 Arquitetura

A Figura 16 explicita a arquitetura geral para disponibilização dos modelos. Ela é composta por cinco elementos, um deles é o usuário que irá fazer uso da Crevettic para realizar a contagem de animais. A própria plataforma, que como citado anteriormente é um sistema com um conjunto de módulos, onde serão implementadas novas funcionalidades para que seja possível consumir informações da API MIC. A API MIC, que será construída no padrão Transferência Representacional de Estado (REST), é a responsável por gerenciar a conexão entre a Crevettic e os modelos de predição. E, por fim, há o banco de dados, onde são armazenados todos os dados da plataforma.

Figura 16 – Arquitetura geral da solução



Fonte: autoria própria (2021)

O MIC foi desenvolvido utilizando a linguagem *python* e já está disponível na versão 1.0. Dentre as tecnologias e técnicas aplicadas, pode-se citar o gerenciamento de versões, que foi realizado via *GitHub*, a utilização do *framework Swagger* para realizar a documentação das rotas, além da criação de testes automatizados com *pytest* para facilitar a manutenção, implementação e validação do módulo. Nessa versão foram implementadas rotas para *download* e *upload* de imagens, em diversos formatos, acesso aos modelos de predição MAIV; e ao de contagem de tilápias. Na figura 17 é apresentada a tela inicial do *Swagger*, demonstrando as rotas desenvolvidas até o momento.

Figura 17 – MIC - Exibição da tela Swagger



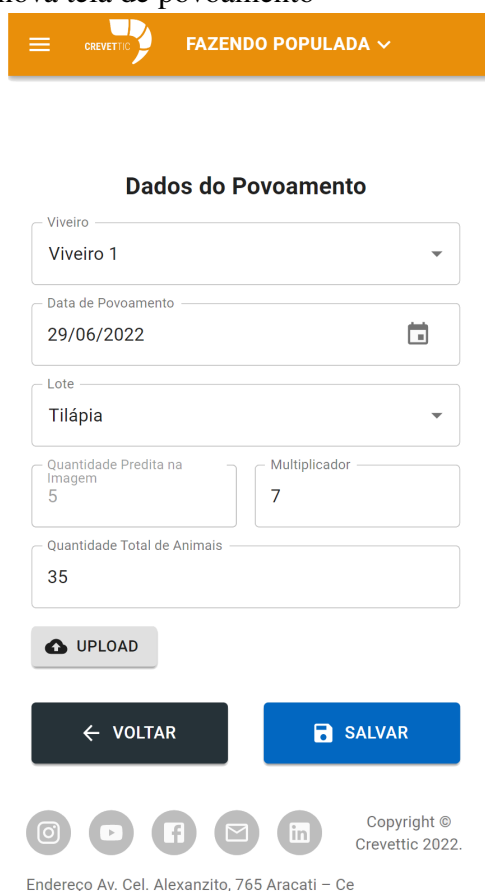
Fonte: autoria própria (2021)

Uma nova versão da tela de povoamento foi desenvolvida para permitir que usuário envie uma imagem e o sistema retorne a predição, se a imagem está ou não em confor-



midade, bem como a quantidade de animais. Há, ainda, a possibilidade de informar o multiplicador (quantidade de vezes que foi utilizado o recipiente), caso ele esteja usando o método de contagem por amostragem, para assim, o sistema calcular o total de animais do povoamento. Essa implementação é muito importante, pois comprova que, mesmo a plataforma sendo desenvolvida com linguagem e tecnologias diferentes, é possível realizar a comunicação com o MIC. Na Figura 18 expõe-se a nova versão da tela de povoamento.

Figura 18 – Proposta de nova tela de povoamento



A interface de usuário para a tela de povoamento, apresentando um cabeçalho laranja com o menu, logo e título. O formulário principal, intitulado 'Dados do Povoamento', contém campos para selecionar o viveiro, a data de povoamento, o lote, a quantidade predita na imagem, o multiplicador e a quantidade total de animais. Abaixo do formulário, há um botão 'UPLOAD' e dois botões de ação: 'VOLTAR' e 'SALVAR'. O rodapé inclui ícones de redes sociais, informações de copyright e o endereço da empresa.

**Cabeçalho:**

- Menu (hambúrguer)
- CREVETIC
- FAZENDO POPULADA

**Dados do Povoamento**

Viveiro: Viveiro 1

Data de Povoamento: 29/06/2022

Lote: Tilápia

Quantidade Predita na Imagem: 5

Multiplicador: 7

Quantidade Total de Animais: 35

**Botões de Ação:**

- UPLOAD
- VOLTAR
- SALVAR

**Rodapé:**

- Ícones de redes sociais: Instagram, YouTube, Facebook, Email, LinkedIn
- Copyright © Crevettic 2022.
- Endereço: Av. Cel. Alexanzito, 765 Aracati – Ce

Fonte: autoria própria (2021)

## 6 RESULTADOS

Para melhor entendimento da dissertação, convém apresentar os resultados dos modelos de forma separada.

### 6.1 MAIV

Nesta seção, expõe-se e discute-se os resultados obtidos para os modelos de detecção de *outliers*. Inicia-se ressaltando que todos os modelos obtiveram bons resultados, então, para uma melhor análise optou-se por usar as seguintes métricas:

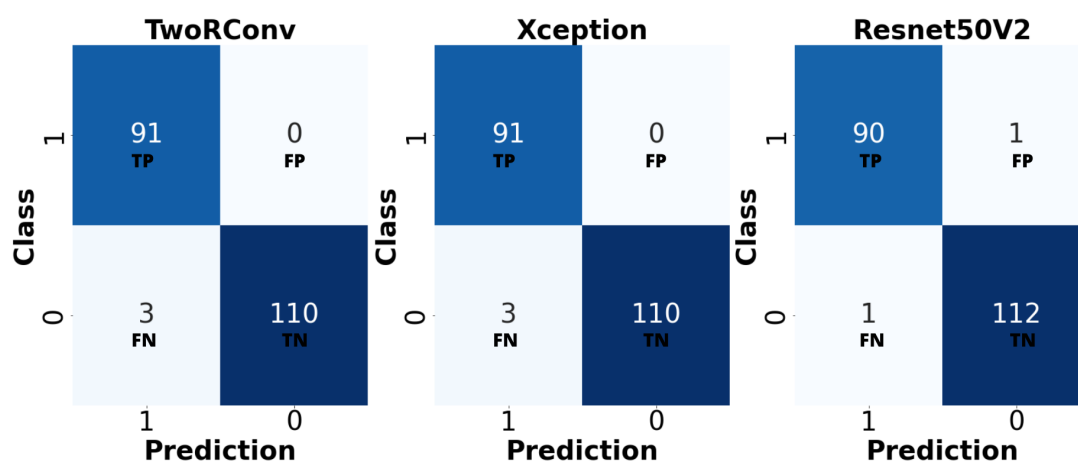
$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6.1)$$

$$precision = \frac{TP}{TP + FP} \quad (6.2)$$

$$recall = \frac{TP}{TP + FN} \quad (6.3)$$

Os valores onde TP, TN, FP, FN correspondem respectivamente aos verdadeiros positivos, verdadeiros negativos, falsos positivos, falsos negativos. Onde os valores TP são os itens que são verdadeiros e o modelo classificou corretamente como verdadeiros, TN que são negativos e foram classificados corretamente como negativos, FN são valores positivos que foram erroneamente classificados negativos, e por fim o FP que são os negativos que foram classificados erroneamente como positivos. Os valores retornados pelos modelos são exibidos nas matrizes de confusão, que é o mapa dos valores TP, TN, FP e FN, sendo apresentados na Figura 19, sendo que a classe conformidade é representada pelo valor 1 e a não conformidade por 0.

Figura 19 – Matriz Confusão - Modelos.



Fonte: autoria própria (2021)

Os valores referentes às métricas que cada modelo obteve estão dispostos na Tabela 4. Todos os modelos apresentaram uma *accuracy* de 99%, e apenas o modelo *Resnet50V2* não obteve o *recall* de 100%, sendo o único que classificou erroneamente uma imagem em conformidade como em não conformidade, conforme sua matriz confusão na Figura 19. O modelo *Resnet50V2* obteve a *precision* de 99% sendo a maior entre os três modelos.

Para o problema em questão, destaca-se que é preferível o modelo que minimize a quantidade de falsos negativos para, assim, minimizar a quantidade de imagens inválidas enviadas ao modelo de contagem. Dessa forma, o *Xception* é considerado o melhor modelo, pois ele obteve a *precision* mais elevada e o menor número de FN. Seguindo assim o padrão de trabalhos como Agouri *et al.* (2022) e Mahdianpari *et al.* (2018), em que o modelo *Xception* obteve melhor desempenho em comparação com outros modelos de CNN.

Metric	TwoRConv	Xception	Resnet50V2
accuracy	<b>0.99</b>	<b>0.99</b>	<b>0.99</b>
precision	0.97	0.97	<b>0.99</b>
recall	<b>1</b>	<b>1</b>	0.99

Tabela 4 – Resultados obtidos.

## 6.2 Modelos de Contagem

### 6.2.1 Métricas

Os modelos de contagem foram avaliados por três métricas, sendo elas mAP, *accuracy* e RMSE. A primeira é baseada em diversos fatores como *precision*, *recall*, posicionamento da caixa e outros, mas a segunda e a terceira são relacionadas com a quantidade de caixas delimitadoras informadas no arquivo e a quantidade retornada pela predição do modelo.

Conforme Blasco *et al.* (2013), o *Mean Absolute Percentage Error* (MAPE) é um dos mecanismos de avaliação de precisão mais utilizados, e tem características importantes e desejáveis, como confiabilidade, medida sem unidade e facilidade de interpretação. Diversos trabalhos analisados na seção 3.2, também analisaram essa métrica para se medir a *accuracy* dos modelos. A formula é exposta na próxima equação, sendo  $n$  o número de itens analisados,  $\hat{y}$  a predição do modelo, e  $y$  o valor real. Deduzindo o seu valor de 100, obtem-se o valor da *accuracy*.

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \frac{|\hat{y}_t - y_t|}{y_t} \quad (6.4)$$

O segundo parâmetro utilizado foi RMSE, que é uma métrica estatística padrão amplamente escolhida para medir o desempenho de modelos. Tal parâmetro penaliza os erros encontrados, pois eles são elevados ao quadrado antes de ser calculada a média, logo, erros maiores terão pesos maiores (CHAI; DRAXLER, 2014). O trabalho analisado (KAEWCHOTE *et al.*, 2018) também usou a métrica como medida de desempenho. As variáveis utilizadas são as mesmas da equação anterior, e o seu cálculo pode ser observado na equação a seguir.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (\hat{y}_t - y_t)^2} \quad (6.5)$$

E por fim, temos a métrica mAP, que é o valor médio de *Average Precision* (AP) de todas as classes presentes no modelo, sendo a métrica padrão de competição do modelo *PascalVOC* (BOCHKOVSKIY *et al.*, 2020). AP é o valor médio de 11 pontos na curva *precision/recall* para cada limiar possível (cada probabilidade de detecção), isso numa

mesma classe (EVERINGHAM *et al.*, 2010). Essa métrica é constantemente escolhida para avaliação de modelos YOLO, conforme os trabalhos (WU *et al.*, 2020; ABINAYA *et al.*, 2022; YU; ZHANG, 2021). Na equação 6.6, é apresentado o cálculo para se obter o mAP, sendo que  $C$  é a quantidade de classes.

$$\text{mAP} = \frac{\sum_{i=1}^C AP_i}{C} \quad (6.6)$$

### 6.2.2 Gerações

No decorrer da criação dos modelos de predição para contagem, foram selecionadas as configurações definidas na seção 5.3, sendo realizadas duas gerações de treinamento e validações: a primeira com objetivo de analisar a viabilidade do *dataset* e do modelo da rede; e a segunda com objetivo de refinamento dos resultados. Na primeira geração, foram utilizadas as imagens em conformidade que foram pré-processadas dos subconjuntos tilapia\_002 e tilapia\_003, dividindo em 80% para treinamento e 20% para validação, totalizando 770 imagens. Na primeira geração, o subconjunto tilapia\_003 possuía apenas 587 imagens pré-processadas.

Para a segunda geração, foram utilizadas as imagens dos subconjuntos tilapia\_002, tilapia\_003 e tilapia\_004 e 152 imagens de vazio\_001. Optou-se também por incluir 152 imagens sem nenhum animal, pois imagens sem caixas delimitadoras servem para aprendizado de padrões e redução do número de falsos positivos (ADIBHATLA *et al.*, 2021; ETIENNE *et al.*, 2021). Dividiu-se essa amostra em 80% para treinamento e 20% para validação, totalizando 926 imagens. O grande diferencial aqui é que não foram descartadas as imagens de tilapia\_002 que continham mais de 33 animais e tilapia\_003 já estava com 685 imagens pré-processadas. O descarte de algumas imagens de tilapia\_002 se deve pelo fato da sobreposição dos animais, conforme explicado a seguir.

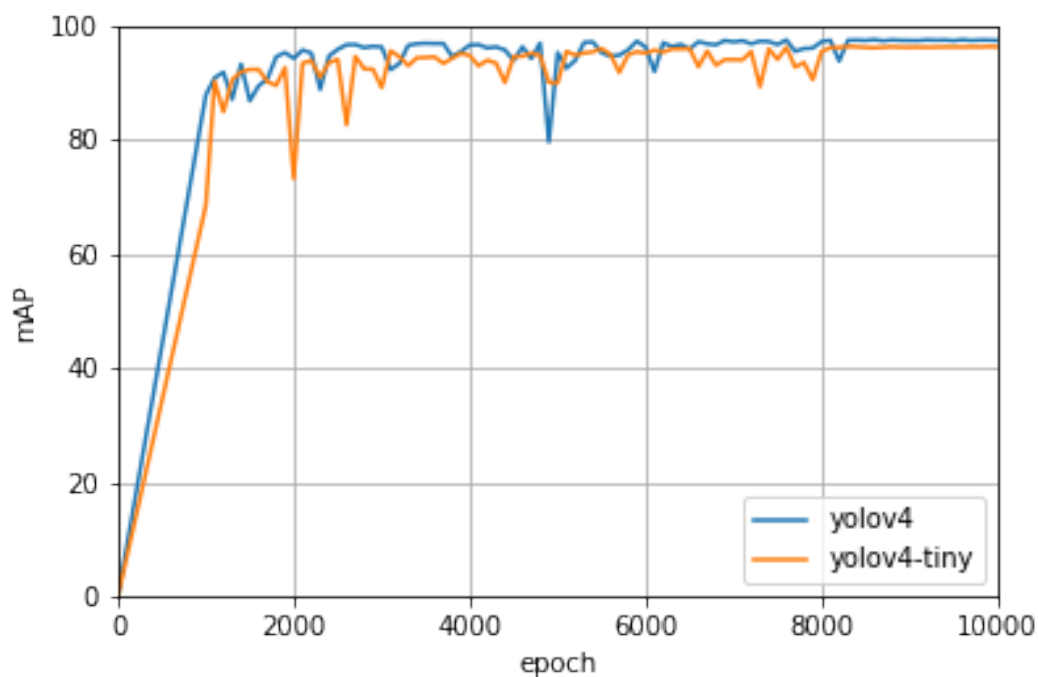
### 6.2.3 Primeira Geração de Modelos

No decorrer desta seção, são apresentados os resultados da primeira rodada de treinamento e validação dos modelos de contagem de animais, utilizando as redes *YOLOv4* e *YOLOv4-tiny*.

As Figuras 20 e 21 mostram o mAP e *loss* obtidos durante o processo de treinamento.

Observa-se que a precisão do modelo aumenta gradualmente à medida que o número de iterações aumenta, mas nota-se também que o valor da perda de treinamento do modelo diminuiu. Nas primeiras duas mil iterações, a eficiência de aprendizado do modelo se mostrou maior e a curva de aprendizagem convergiu mais rápido. Para ambos os modelos, tanto a curva de aprendizagem, representada pelo mAP, e a *loss* diminuem as oscilações por volta das 9.000 iterações. O treinamento com *YOLOv4* foi finalizado com *map* de 97,27% e a *YOLOv4-tiny* com 96,14%.

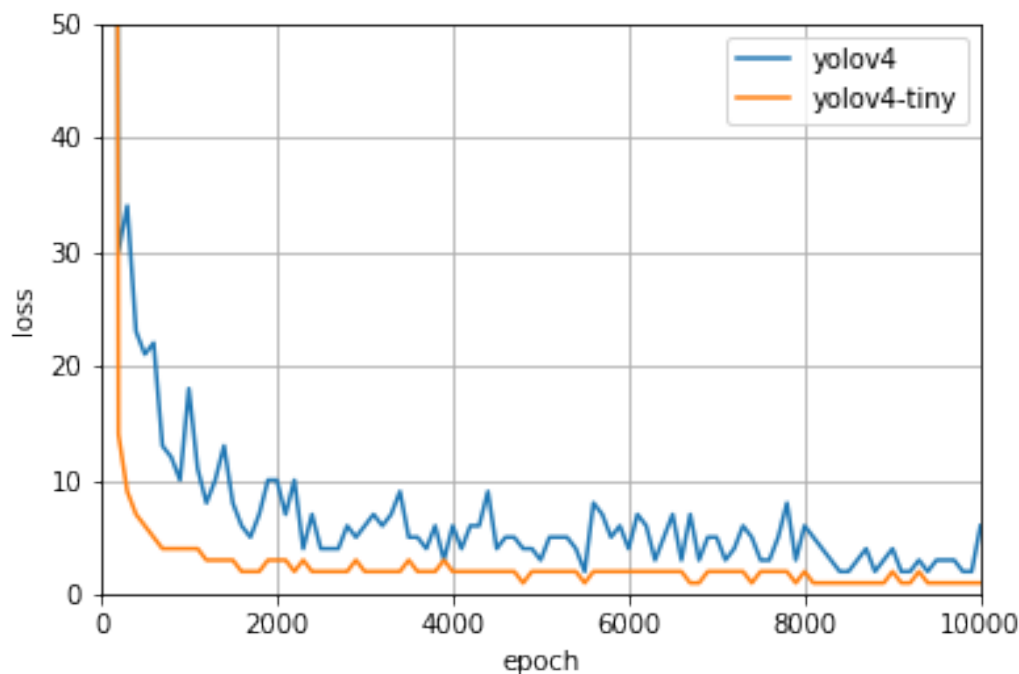
Figura 20 – Gráfico apresentando o *mAP* do modelo de treinamento das redes *YOLOv4* e *YOLOv4-tiny* da primeira geração, com as configurações informadas na seção 5.3



Fonte: autoria própria (2021)

Conforme os trabalhos (WU *et al.*, 2020; YANG *et al.*, 2021), e a documentação do *framework*, optou-se por pausar o treinamento quando, mesmo com a continuidade das iterações, não se apresentou aumento significativo do mAP e nem redução do *loss*. De modo que pode-se identificar que o modelo não está sofrendo de *overfitting*.

Figura 21 – Gráfico apresentando a *loss* do modelo de treinamento das redes *YOLOv4* e *YOLOv4-tiny* da primeira geração, com as configurações informadas na seção 5.3



Fonte: autoria própria (2021)

O mAP é calculado com base em um limiar de ativação definido pelo *Intersection over Union* (IoU), que é a confiança do modelo em determinar a caixa delimitadora. Foi utilizada a configuração de *threshold* IoU com valor de 0.5, conforme os trabalhos Bazzaza *et al.* (2022) e Yang *et al.* (2021).

Na tabela 5 são expostos o resultados dos modelos, sendo possível perceber que a rede *YOLOv4* obteve o melhor desempenho em todos os itens. Esses dados fazem ela ser considerada a melhor opção desta geração, por ser a superior, e, por isso, faz-se uma análise aprofundada dos resultados dela. O *mAP* e a *accuracy* obtiveram um desempenho satisfatório, porém deve-se atentar ao *RMSE*.

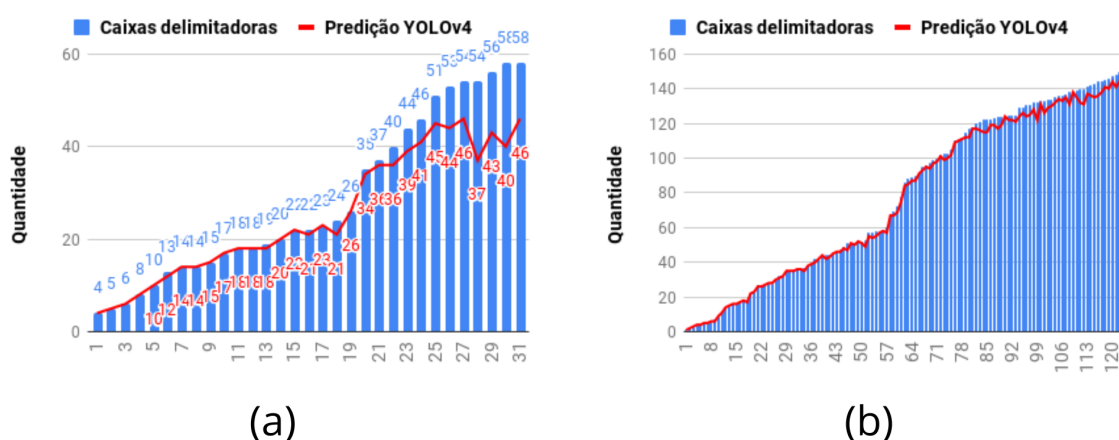
	Validação		
	mAP	accuracy	RMSE
YOLOv4	<b>97.27</b>	<b>96.64</b>	<b>4.12</b>
YOLOv4-tiny	96.12	96.03	4.44

Tabela 5 – Resultados dos modelos de contagem da primeira geração.

Para compreensão do valor da métrica *RMSE*, deve-se observar a quantidade de

quadros delimitadores e a quantidade de animais prevista. Na figura 22, as colunas representam a quantidade de box em cada imagem e a linha vermelha o retorno do modelo. Pode-se perceber que, para o conjunto *tilápia\_002*, a partir do item 22, que possui 37 caixas, o modelo começa a apresentar dificuldade em prever a quantidade aproximada, mas, para o segundo subconjunto, o modelo se manteve estável ao longo de todos os itens.

Figura 22 – Gráfico apresenta a relação entre a quantidade de caixas delimitadoras e quantidade de animais previstos do modelo *YOLOv4* primeira geração.



Fonte: autoria própria (2021)

O gráfico na Figura 23 apresenta o *absolute error*, que é módulo da diferença entre a predição e a quantidade de caixas, dividido pela quantidade de caixas, dos itens do conjunto de testes. O valor apresentado por item é usado no somatório do *MAPE*. Já no *RMSE*, a diferença entre as quantidades de caixas e as predições são elevadas ao quadrado, potencializando o erro das predições que se afastem do valor correto, justamente com o objetivo de penalizar predições erradas. Na figura, a linha azul representa os itens de *tilapia\_002* e a verde a *tilapia\_003*. Logo, é possível identificar que, mesmo o conjunto *tilapia\_003* mantendo o seu valor estável, o conjunto *tilapia\_002* apresentou valores muito discrepantes a partir do item 22, o que acabou aumentando o valor de *RMSE* para todo conjunto de testes. Caso os valores fossem analisados de forma separada, o conjunto de testes de *tilapia\_002* teria: *accuracy*: 92.97; e *RMSE*: 6.18, enquanto o *tilapia\_003* teria: *accuracy*: 97.56 e *RMSE*: 3.42.



Figura 23 – *Absolute Error* - Conjunto de Validação *YOLOv4* primeira geração.



Fonte: Autoria Própria (2021)

Para entender o problema enfrentado pelo modelo, primeiro faz-se necessário citar as especificações dos subconjuntos, conforme a tabela 2. O primeiro subconjunto foi registrado utilizando uma placa Petri de 13 cm de diâmetro, enquanto o segundo com uma de 25 cm de diâmetro. Essa diferença faz com que a área disponível para a disposição dos animais seja 3.7 vezes maior no subconjunto *tilapia\_003*. Essa diferença implica diretamente na sobreposição dos animais, um fenômeno que dificulta e em alguns casos, inviabiliza a contagem dos mesmos, tendo em vista que não é possível identificá-los quando estão agrupados um sobre o outro. É importante frisar que essa dificuldade também foi relatada pelos pesquisadores Boksuwan *et al.* (2018), Labuguen *et al.* (2012) e Hernández-Ontiveros *et al.* (2018).

Um exemplo desse processo pode ser observado na Figura 24, pois, na parte inferior da primeira imagem, os animais estão amontoados de tal modo que fica inviável o processo de contagem por meio visual. Enquanto que, na segunda imagem, o recipiente é bem maior e os animais se encontram distribuídos de maneira que é possível realizar a contagem. Vale ressaltar que ambas as imagens possuem 58 animais, porém são de subconjuntos diferentes, o que exemplifica que o tamanho do recipiente faz total diferença na dispersão dos animais.

Figura 24 – Imagens com 58 animais dos subconjuntos tilapia\_002 e tilapia\_003



**subconjunto: tilapia\_002**  
**vd\_003\_1\_0000175.jpg**

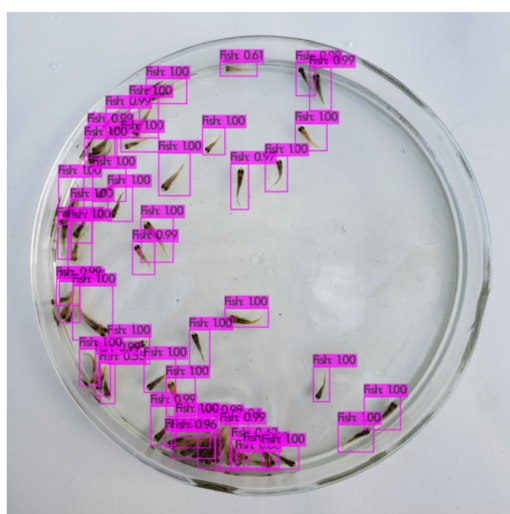


**subconjunto: tilapia\_003**  
**vd\_005\_1\_0000462.jpg**

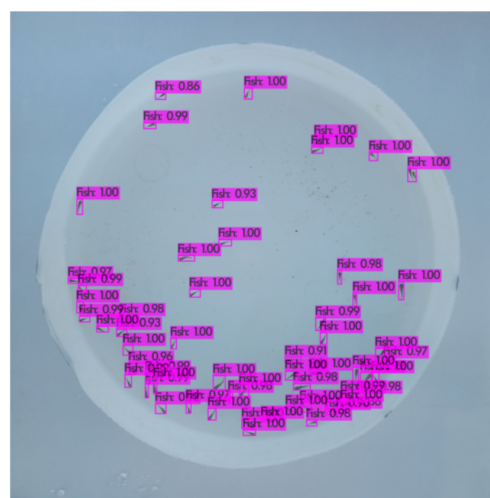
Fonte: autoria própria (2021)

Na Figura 25, demonstra-se a saída do modelo para as duas imagens anteriores, sendo que o sistema informou, respectivamente, 40 animais para primeira imagem e 57 para segunda. Dessa forma, evidencia-se que a dificuldade do modelo não está em identificar animais, mas sim em identificar animais sobrepostos.

Figura 25 – Imagens com 58 animais dos subconjuntos tilapia\_002 e tilapia\_003 retornadas pelo modelo de contagem YOLOv4 primeira geração.



**subconjunto: tilapia\_002**  
**vd\_003\_1\_0000175**



**subconjunto: tilapia\_003**  
**vd\_005\_1\_0000462**

Fonte: autoria própria (2021)

Logo, de acordo com os dados analisados, optou-se que, na geração de modelos da segunda etapa, não seriam utilizados os arquivos de tilapia\_002 que contassem com mais de 33 animais, pois percebe-se que imagens em que há um elevado nível de sobreposição, podem ocorrer *outliers* e, frente ao cenário de dúvidas, é melhor descartá-las.

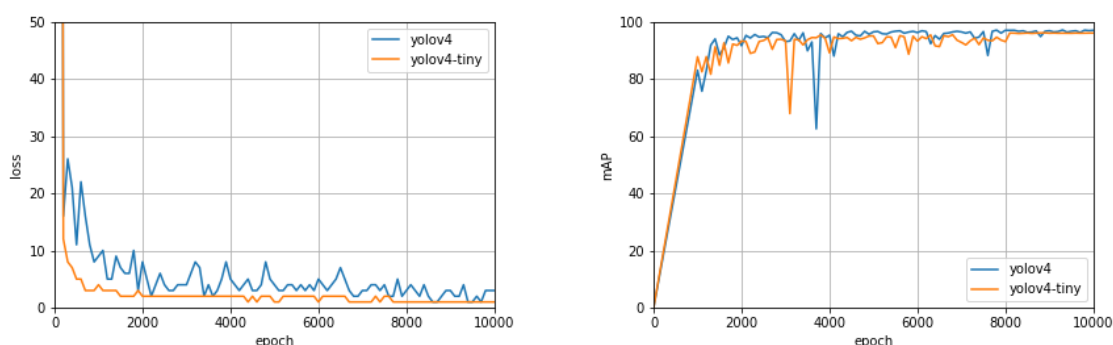
#### 6.2.4 Segunda Geração de Modelos

Com objetivo de criar um modelo mais preciso na segunda geração, foram escolhidas, para treinamento e validação as seguintes imagens: 102 imagens de tilapia\_002; 685 de tilapia\_003, que é o total de pré-processadas até o momento para este subconjunto e 63 de tilapia\_004. Dessa forma, o modelo em questão utilizou um total de 926 imagens.

Para treinamento dos modelos utilizou-se a mesma configuração da seção anterior. Na figura 26 é apresentado os gráficos de *loss* e *mAp* da etapa de treinamento, semelhante a versão anterior pode-se perceber que a precisão do modelo aumenta a medida que as iterações aumentam, se estabilizando por volta de nove mil. De modo que é pausado o treinamento antes de ocorrer *overfitting*. Finalizando o treinamento da *YOLOv4* com *mAP* de 96,94% e a *YOLOv4-tiny* de 95,98%.

Para treinamento dos modelos, utilizou-se a mesma configuração explicitada da seção anterior. Na Figura 26, expõe-se os gráficos de *loss* e *mAp* da etapa de treinamento e semelhante à versão anterior, pode-se perceber que a precisão do modelo aumenta à medida em que as iterações também aumentam, se estabilizando por volta de nove mil. Dessa forma, pode-se afirmar que o modelo tem o treinamento pausado antes de ocorrer *overfitting*, o que permite a conclusão da *YOLOv4* com *mAP* de 96,94% e a *YOLOv4-tiny* de 95,98%.

Figura 26 – Gráficos apresentando *loss* e *mAP* do modelo de treinamento das redes *YOLOv4* e *YOLOv4-tiny* segunda geração, com as configurações informadas na seção 5.3



Fonte: autoria própria (2021)

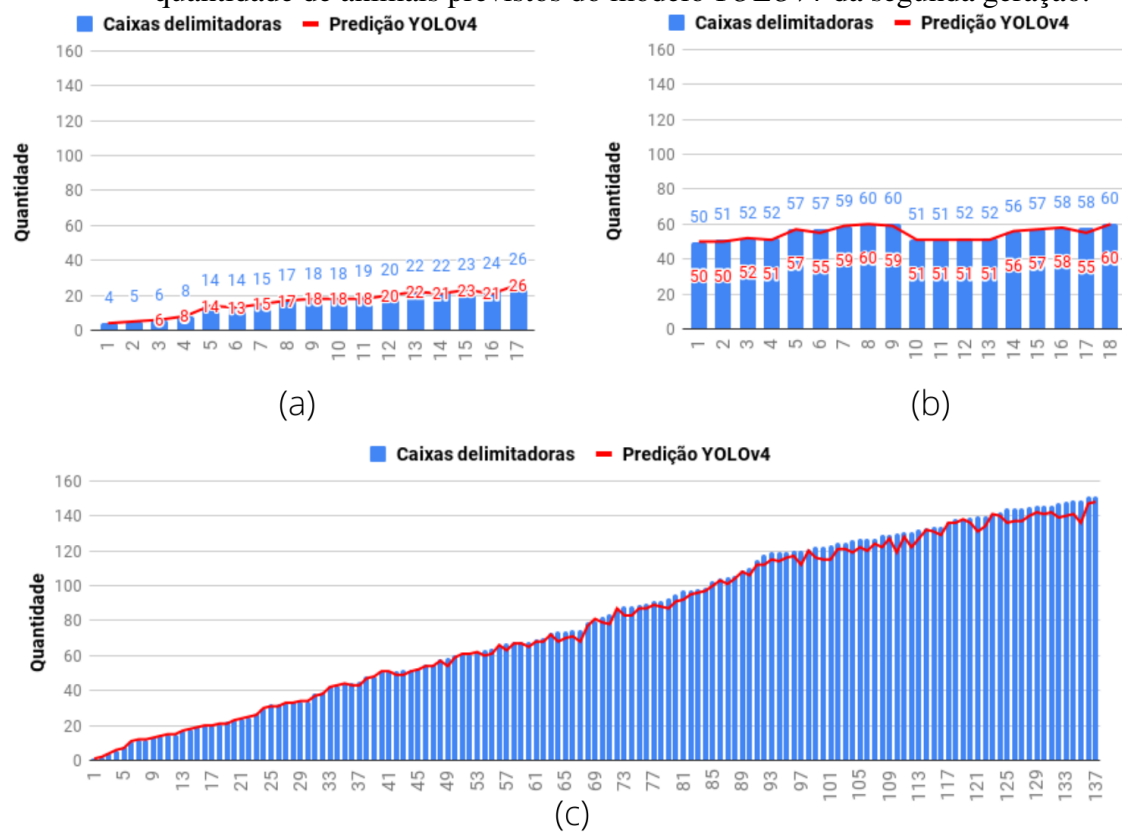
Em função da limitação do *hardware*, os modelos só puderam ser treinados com resolução de 416x416, porém, na etapa de validação, esse valor pode ser alterado, permitindo a comparação do desempenho (AL-QUDAH; SUEN, 2019). A tabela 6 compara os resultados obtidos com resolução 416 x 416 e 608 x 608. Vale ressaltar que, para cálculo da *accuracy* e *RMSE*, não foram consideradas as imagens com zero animais, pois esse uso poderia comprometer a comparação com modelos da geração anterior. Ambos os modelos tiveram o seu desempenho melhorado ao alterar a resolução de entrada das imagens. Além disso, repetindo o padrão da primeira geração, o modelo *YOLOv4* obteve um melhor desempenho, independente da resolução. Logo, o melhor modelo desenvolvido foi o *YOLOv4* com resolução de entrada 608x608.

	Validação					
	416x416			608x608		
	mAP	accuracy	RMSE	mAP	accuracy	RMSE
YOLOv4	<b>96.94</b>	<b>97.50</b>	<b>3.48</b>	<b>98.50</b>	<b>98.70</b>	<b>2.07</b>
YOLOv4-tiny	95.98	96.77	3.95	97.43	97.44	2.92

Tabela 6 – Resultados dos modelos de contagem da segunda geração.

Na figura 27 apresenta-se a quantidade de caixas delimitadoras (coluna azul) das imagens e a quantidade prevista pelo modelo (linha vermelha) *YOLOv4*, sendo a ordem tilapia\_002, tilapia\_003 e tilapia\_004. Observa-se que, diferente do modelo da primeira geração, neste a linha vermelha está sempre próxima da quantidade esperada e essa diferença reflete diretamente no valor do RMSE, que caiu praticamente pela metade em relação ao modelo anterior.

Figura 27 – Gráfico apresentando a relação entre a quantidade de caixas delimitadoras e quantidade de animais previstos do modelo *YOLOv4* da segunda geração.



Fonte: autoria própria (2021)

O *absolute error* do conjunto de validação do modelo *YOLOv4* de segunda geração aparece na figura 28. Ao comparar a mesma métrica com modelo da primeira geração, verifica-se um desempenho nitidamente melhor, tendo em vista que os valores apresentaram uma baixa variância, se mantendo quase que exclusivamente abaixo de dez, com exceção de dois casos. Essa métrica, juntamente com os dados apresentados anteriormente, comprovam efetivamente a superioridade da segunda geração em relação à primeira.

Figura 28 – *Absolute Error* - Conjunto de Validação *YOLOv4* da segunda geração.



Fonte: autoria própria (2021)

Na Figura 29, registra-se uma predição do modelo *YOLOv4* segunda geração para imagens com 20, 58 e 109 animais, onde o modelo informou, respectivamente 20, 58 e 108 animais. Em cada uma das imagens retornadas, foram utilizados *smartphones* e recipientes diferentes, apresentando, assim uma heterogeneidade do processo de criação dos subconjuntos. Portanto, pode-se concluir que, mesmo diante de um ambiente diversificado e em quantidade diferentes de animais o modelo apresentou excelentes resultados.

Figura 29 – Imagens processadas *YOLOv4* da primeira geração.



vd\_003\_1\_0000061

vd\_006\_1\_0000076

vd\_005\_1\_0001061

Fonte: autoria própria (2021)

Para corroborar os resultados apresentados anteriormente, os modelos *YOLOv4* e *YOLOv4-tiny*, com resolução de entrada 608x608 foram testados na predição do subconjunto *tilapia\_005*. Esse subconjunto não foi utilizado durante as etapas de treinamento e validação, mas possui 40 imagens e a quantidade de animais varia entre 51 e 60, tendo sido gerada através do *smartphone Samsung A51*, um aparelho não utilizado em nenhum outro subconjunto. Esse teste, apesar da limitação quanto a quantidade de animais e à quantidade de amostras, apresenta uma experimentação real do modelo e, conforme a tabela 7, os resultados obtidos são elevados e satisfatórios, caracterizando, novamente, o *Yolov4* com os melhores resultados, alcançando um mAP de 98,63% e *accuracy* de 98,66%.

	tilapia_006		
	mAP	accuracy	RMSE
YOLOv4	<b>99.63</b>	<b>98.66</b>	<b>1.07</b>
YOLOv4-tiny	98.13	97.15	2.04

Tabela 7 – Resultados dos modelos de contagem da segunda geração para *tilapia\_005*

Diante dos dados apresentados, pode-se concluir que o modelo *YOLOv4* da segunda geração, com resolução de entrada 608x608, obteve os melhores resultados em relação aos demais modelos e é uma opção válida na construção de um conjunto de soluções para predição de contagem de animais. Ressalta-se, inclusive, que tal modelo demonstrou um excelente resultado diante de imagens inéditas. Entretanto, é preciso pontuar que, apesar dos bons resultados, pretende-se continuar aprimorando o mesmo, sobretudo com o treinamento e validação diante de imagens com maior quantidade de animais e com animais de tamanhos diferentes, pois, acredita-se que, assim, será possível desenvolver um modelo que possa atender a uma quantidade cada vez maior de situações.

## 7 CONCLUSÃO

O presente trabalho apresentou a metodologia e registrou a criação da primeira versão do Vivarium, além de apresentar o modelo MAIV, que tem 99% de precisão em definir se uma imagem está ou não em conformidade com as especificações definidas pela metodologia. Além da criação de um modelo de contagem de tilápias, que obteve mAP de 98.50% e *accuracy* de 98.70%.

Registra-se, também, o desenvolvimento de toda uma estrutura para que os modelos possam ser disponibilizados para o público, sendo que tais modelos apresentados e o Vivarium são os primeiros passos em direção à construção de um módulo de contagem escalável e de fácil acesso, principalmente para o pequeno e médio produtor da aquicultura.

Verifica-se como limitações do trabalho as especificidades dos *datasets* utilizados para treinamento e a avaliação dos modelos, devido a limitações referentes a quantidade, espécie e tamanho dos animais. Logo, é válido em trabalhos futuros realizar uma análise da predição com uma quantidade maior de animais e em tamanhos diferentes, podendo inclusive ser retreinado os modelos caso seja necessário.

Como trabalhos futuros, espera-se continuar o desenvolvimento do Vivarium, a partir da inclusão de novos subconjuntos, abrangendo uma maior quantidade de animais, outras fases de cultivo, além de outras espécies de animais aquáticos, como o camarão marinho *Litopenaeus vannamei*, bem como outras espécies de peixes.

Pretende-se também agregar uma nova funcionalidade ao o MAIV, que é identificação de imagens com alto nível de sobreposição de animais como *outliers*. Para assim caso o usuário final envie uma imagem sem considerar a distribuição dos animais no recipiente o mesmo possa ser alertado.

Quanto ao modelo de avaliação e contagem de imagens, pretende-se atualizá-lo à medida que novos subconjuntos forem sendo inseridos e ajustes sejam realizados para otimizar os resultados. É planejado, também, criar novos modelos aplicando apenas técnicas de visão computacional, como, por exemplo segmentação, e, em seguida, avaliar se é possível obter melhor desempenho em relação ao custo computacional e aos resultados obtidos. Assim, firma-se, como próximo passo do projeto, o aprimoramento dos modelos para que a contagem possa ser realizada também por vídeo, garantindo maior precisão e mitigando o problema da sobreposição de animais.



## 8 PRODUTIVIDADE ACADÊMICA E CONQUISTAS

No decorrer do mestrado foram desenvolvidos diversos trabalhos acadêmicos, participações em eventos de divulgação científica, tecnológica e de inovação. Segue alguns exemplos a seguir.

### 8.1 Trabalhos Publicados

*Semantic strategies in ubiquitous music: Deploying the sound sphere ecology in transitional settings*, na revista *Helion* (KELLER *et al.*, 2020).

A Ferramenta SoundSphere no Contexto da Exploração e da Criação Sonora: Um Estudo de Caso de Música Ubíqua dentro da Sala de Aula, em *Anais do X Workshop de Música Ubíqua* (SIMURRA *et al.*, 2020).

Redes Neurais Convolucionais Aplicadas no Diagnóstico do Câncer de Pulmão, *Anais do XIV Encontro Unificado de Computação do Piauí* (BESSA *et al.*, 2021).

Um Sistema Inteligente de Monitoramento e Avaliação da Saúde do Camarão, *anais do XXVII SEMIC UFERSA* (SOUZA *et al.*, 2021).

Gastrossônica: Desenvolvimento do suporte ubimus para atividades musicais gastrôssônicas, nos *anais do 11º Workshop on Ubiquitous Music* (FREITAS *et al.*, 2021).

*Image Filtering Model Based on Convolutional Neural Networks for Automatic Counting of Post-larvae in Aquaculture*, no *11th Euro American Conference on Telematics and Information Systems* (BESSA *et al.*, 2022). Vale ressaltar que essa publicação obteve a melhor pontuação no processo de revisão, sendo considerado o melhor artigo do evento. Nela foi apresentado o método de criação do *dataset* Vivarium e o MAIV.

### 8.2 Trabalhos apresentados

Redes Neurais Convolucionais Aplicadas no Diagnóstico do Câncer de Pulmão, *XIV Encontro Unificado de Computação do Piauí*.

Linkfish - plataforma inteligente de comercialização de produtos aquícolas, *19º Congresso de Engenharia de Aquicultura*.

Crevettic: uma plataforma de business intelligence para gestão da aquicultura, no *Fórum de Inovação em Agronegócios*.

*Image Filtering Model Based on Convolutional Neural Networks for Automatic*

*Counting of Post-larvae in Aquaculture, no 11th Euro American Conference on Telematics and Information Systems.*

### **8.3 Capítulo de livros**

Linkfish - plataforma inteligente de comercialização de produtos aquícolas (BARBOSA *et al.*, 2021).

Crevettic: uma plataforma de *business intelligence* para gestão da aquicultura (NETO *et al.*, 2021)

## REFERÊNCIAS

- ABCC. **Produção Brasileira de Camarão Marinho Cultivado por Estado: Dados reais de 2015 a 2019**. 2021. Disponível em: <<https://abccam.com.br/wp-content/uploads/2020/10/Producao-de-Camarao-2020.pdf>>.
- ABINAYA, N.; SUSAN, D.; SIDHARTHAN, R. K. Deep learning-based segmental analysis of fish for biomass estimation in an occulted environment. **Computers and Electronics in Agriculture**, Elsevier, v. 197, p. 106985, 2022.
- ADIBHATLA, V. A.; HUANG, Y.-C.; CHANG, M.-C.; KUO, H.-C.; UTEKAR, A.; CHIH, H.-C.; ABBOD, M. F.; SHIEH, J.-S. Unsupervised anomaly detection in printed circuit boards through student–teacher feature pyramid matching. **Electronics**, MDPI, v. 10, n. 24, p. 3177, 2021.
- AGOURI, H. E.; AZIZI, M.; ATTAR, H. E.; KHANNOUSSI, M. E.; IBRAHIMI, A.; KABBAJ, R.; KADIRI, H.; BEKARSABEIN, S.; ECHCHARIF, S.; MOUNJID, C. *et al.* Assessment of deep learning algorithms to predict histopathological diagnosis of breast cancer: first moroccan prospective study on a private dataset. **BMC Research Notes**, BioMed Central, v. 15, n. 1, p. 1–7, 2022.
- AL-QUDAH, R.; SUEN, C. Y. Enhancing yolo deep networks for the detection of license plates in complex scenes. In: **Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems**. [S.l.: s.n.], 2019. p. 1–6.
- AMÉRICO, M.; AMORIM, B.; CRISTO, G.; VALLE, J. B.; SALGADO, J. S. Práticas de processamento e aproveitamento de subprodutos de pescado para agregação de valor, comercialização e geração de renda para agricultores familiares do norte do espírito santo. **Cadernos de Agroecologia**, v. 8, n. 2, 2013.
- BACKES, A. R.; JUNIOR, J. J. d. M. S. **Introdução à visão computacional usando Matlab**. [S.l.]: Alta Books Editora, 2019.
- BALOGLU, O.; LATIFI, S. Q.; NAZHA, A. What is machine learning? **Archives of Disease in Childhood-Education and Practice**, Royal College of Paediatrics and Child Health, 2021.
- BARBOSA, V. N.; NETO, F. M. M.; LEITE, D. G.; RIBEIRO, M. H. D.; BESSA, W. R. B.; BRAGA, O. C. Linkfish - plataforma inteligente de comercialização de produtos aquícolas. In: VALENÇA, A. R.; SANTOS, P. R. dos; GUZELLA, L. (Ed.). **Pesquisas e Aquicultura: Desenvolvimento Tecnológico Sustentável**. Florianópolis: UFSC, 2021. cap. 3, p. 51–68. ISBN 978-85-8328-074-3.
- BAZZAZA, T.; CHEN, Z.; PRABHA, S.; TOHIDYPOUR, H. R.; WANG, Y.; POURAZAD, M. T.; NASIOPOULOS, P.; LEUNG, V. C. Automatic street parking space detection using visual information and convolutional neural networks. In: **2022 IEEE International Conference on Consumer Electronics (ICCE)**. [S.l.: s.n.], 2022. p. 01–02.
- BESSA, W. R. B.; BARBOSA, V.; LEITE, D.; NETO, F.; SANTOS, V.; SILVA, T.; ARAÚJO, G.; MOREIRA, M.; BRAGA, O. Image filtering model based on convolutional neural networks for automatic counting of post-larvae in aquaculture. In: **Proceedings of the 12th Euro-American Conference on Telematics and Information Systems**. [S.l.: s.n.], 2022.

BESSA, W. R. B.; NASCIMENTO, R. S. do; FILHO, S. E. A.; SILVA, M. P. D. S.; NETO, F. M. M. Redes neurais convolucionais aplicadas no diagnóstico do câncer de pulmão. In: SBC. **Anais do XIV Encontro Unificado de Computação do Piauí e XI Simpósio de Sistemas de Informação**. [S.l.], 2021. p. 17–24.

BLASCO, B. C.; MORENO, J. J. M.; POL, A. P.; ABAD, A. S. Using the r-mape index as a resistant measure of forecast accuracy. **Psicothema**, Colegio Oficial De Psicólogos Del Principado De Asturias, v. 25, n. 4, p. 500–506, 2013.

BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. **YOLOv4: Optimal Speed and Accuracy of Object Detection**. 2020.

BOKSUWAN, S.; PANAUDOMSUP, S.; CHEYPOCA, T. A prototype system to count nursery pacific white shrimp using image processing. In: IEEE. **2018 18th International Conference on Control, Automation and Systems (ICCAS)**. [S.l.], 2018. p. 1187–1189.

BORTH, M. R.; IACIA, J. C.; PISTORI, H.; RUVIARO, C. F. A visão computacional no agronegócio: Aplicações e direcionamentos. **7º Encontro Científico de Administração, Economia e Contabilidade (ECAECO)**, 2014.

BRAGA, O. C. **CREVETTIC: uma plataforma de business intelligence para gestão de carcinicultura**. 68 p. Dissertação (Mestrado) — Universidade Federal Rural do Semi-Árido, Mossoró, 2020.

CHAI, T.; DRAXLER, R. R. Root mean square error (rmse) or mean absolute error (mae)?—arguments against avoiding rmse in the literature. **Geoscientific model development**, Copernicus GmbH, v. 7, n. 3, p. 1247–1250, 2014.

CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 1251–1258.

Data Science Academy. **Deep Learning Book**. 2021. Disponível em: <<https://www.deeplearningbook.com.br/>>.

DEGADWALA, S.; VYAS, D.; CHAKRABORTY, U.; DIDER, A. R.; BISWAS, H. Yolo-v4 deep learning model for medical face mask detection. In: IEEE. **2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)**. [S.l.], 2021. p. 209–213.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255.

ETIENNE, A.; AHMAD, A.; AGGARWAL, V.; SARASWAT, D. Deep learning-based object detection system for identifying weeds using uas imagery. **Remote Sensing**, MDPI, v. 13, n. 24, p. 5182, 2021.

EVERINGHAM, M.; GOOL, L. V.; WILLIAMS, C. K.; WINN, J.; ZISSERMAN, A. The pascal visual object classes (voc) challenge. **International journal of computer vision**, Springer, v. 88, n. 2, p. 303–338, 2010.

FAO. **The State of World Fisheries and Aquaculture 2020: Sustainability in action.** 2020.

FILHO, M. X. P.; FLORES, R. M. V.; ROCHA, H. S.; SILVA, H. J. T. da; SONODA, D. Y.; CARVALHO, V. B. de; OLIVEIRA, L. de; RODRIGUES, F. L. M. O mercado de peixes da piscicultura no brasil: estudo do segmento de supermercados. **Embrapa Pesca e Aquicultura-Boletim de Pesquisa e Desenvolvimento (INFOTECA-E)**, Palmas: Embrapa Pesca e Aquicultura., 2020.

FREITAS, B.; MEJÍA, C. M. G.; MELO, M. T. de; BESSA, W. R. B.; MESSINA, M.; SIMURRA, I.; KELLER, D. Gastrossônica: Desenvolvimento do suporte ubimus para atividades musicais gastronômicas. In: G-UBIMUS. **Proceedings of the 11th Workshop on Ubiquitous Music (UbiMus 2021)**. [S.l.], 2021. p. 160–171.

GOMES, L. d. C.; ARAUJO-LIMA, C. A. R. M.; ROUBACH, R. Alevino-um termo equivocado na piscicultura brasileira com conseqüências no setor produtivo. **Embrapa Amazônia Ocidental-Artigo em periódico indexado (ALICE)**, Cadernos de Ciência & Tecnologia, Brasília, v. 20, n. 2, p. 353-359, maio, 2003.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.

HENDRYCKS, D.; MAZEIKA, M.; DIETTERICH, T. Deep anomaly detection with outlier exposure. **arXiv preprint arXiv:1812.04606**, 2018.

HERNÁNDEZ-ONTIVEROS, J. M.; INZUNZA-GONZÁLEZ, E.; GARCÍA-GUERRERO, E. E.; LÓPEZ-BONILLA, O. R.; INFANTE-PRIETO, S. O.; CÁRDENAS-VALDEZ, J. R.; TLELO-CUAUTLE, E. Development and implementation of a fish counter by using an embedded system. **Computers and Electronics in Agriculture**, Elsevier, v. 145, p. 53–62, 2018.

JABBAR, H.; KHAN, R. Z. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). **Computer Science, Communication and Instrumentation Devices**, v. 70, 2015.

JIANG, Z.; ZHAO, L.; LI, S.; JIA, Y. Real-time object detection method for embedded devices. In: **computer vision and pattern recognition**. [S.l.: s.n.], 2020.

KAEWCHOTE, J.; JANYONG, S.; LIMPRASERT, W. Image recognition method using local binary pattern and the random forest classifier to count post larvae shrimp. **Agriculture and Natural Resources**, Elsevier, v. 52, n. 4, p. 371–376, 2018.

KELLER, D.; FREITAS, B.; BESSA, W. R. B.; SIMURRA, I.; FARIAS, F. M. de. Semantic strategies in ubiquitous music: Deploying the sound sphere ecology in transitional settings. **Heliyon**, Elsevier, v. 6, n. 9, p. e04843, 2020.

KHANTUWAN, W.; KHIRIPET, N. Live shrimp larvae counting method using co-occurrence color histogram. In: **IEEE. 2012 9th International Conference on**

**Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology.** [S.l.], 2012. p. 1–4.

KOVÁCS, Z. L. **Redes neurais artificiais.** [S.l.]: Editora Livraria da Física, 2002.

KRUMMENAUER, D.; PEIXOTO, S.; CAVALLI, R. O.; POERSCH, L. H.; JR, W. W. Superintensive culture of white shrimp, *litopenaeus vannamei*, in a biofloc technology system in southern brazil at different stocking densities. **Journal of the World Aquaculture Society**, v. 42, n. 5, p. 726–733, 2011. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1749-7345.2011.00507.x>>.

KUSWANTORI, A.; SUESUT, T.; TANGSRIRAT, W.; NUNAK, N. Development of object detection and classification with yolov4 for similar and structural deformed fish. **EUREKA: Physics and Engineering**,(2), p. 154–165, 2022.

LABUGUEN, R.; VOLANTE, E.; CAUSO, A.; BAYOT, R.; PEREN, G.; MACARAIG, R.; LIBATIQUE, N.; TANGONAN, G. Automated fish fry counting and schooling behavior analysis using computer vision. In: IEEE. **2012 IEEE 8th International Colloquium on Signal Processing and its Applications.** [S.l.], 2012. p. 255–260.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Ieee, v. 86, n. 11, p. 2278–2324, 1998.

LI, D.; MIAO, Z.; PENG, F.; WANG, L.; HAO, Y.; WANG, Z.; CHEN, T.; LI, H.; ZHENG, Y. Automatic counting methods in aquaculture: A review. **Journal of the World Aquaculture Society**, Wiley Online Library, v. 52, n. 2, p. 269–283, 2021.

MAHDIANPARI, M.; SALEHI, B.; REZAEI, M.; MOHAMMADIMANESH, F.; ZHANG, Y. Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. **Remote Sensing**, MDPI, v. 10, n. 7, p. 1119, 2018.

MCCARTHY, J. What is artificial intelligence? 2007.

MONKMAN, G. G.; HYDER, K.; KAISER, M. J.; VIDAL, F. P. Using machine vision to estimate fish length from images using regional convolutional neural networks. **Methods in Ecology and Evolution**, Wiley Online Library, v. 10, n. 12, p. 2045–2056, 2019.

MORAIS, E. C. **Reconhecimento de padrões e redes neurais artificiais em predição de estruturas secundárias de proteínas.** Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2010.

NETO, F. M. M.; BRAGA, O. C.; BESSA, W. R. B. crevettic: uma plataforma de business intelligence para gestão da aquicultura. In: PAIVA, A.; VASCONCELOS, E.; VIANA, G. A.; PEREIRA, M. (Ed.). **Inovações em Agronegócios.** Natal: PAX-RN, 2021. cap. 3. ISBN 978-65-00-36320-3.

PARICO, A. I. B.; AHAMED, T. Real time pear fruit detection and counting using yolov4 models and deep sort. **Sensors**, MDPI, v. 21, n. 14, p. 4803, 2021.

Peixe BR. **ANUÁRIO 2021 Peixe BR da Piscicultura.** 2021. Disponível em: <<https://www.peixebr.com.br/anoario-2021/>>.

PEREZ, H.; TAH, J. H. Improving the accuracy of convolutional neural networks by identifying and removing outlier images in datasets using t-sne. **Mathematics**, Multidisciplinary Digital Publishing Institute, v. 8, n. 5, p. 662, 2020.

POUEY, J. L.; CAMARGO, S. Aquicultura-um mercado em expansão. **Current Agricultural Science and Technology**, v. 11, n. 4, 2005.

RAHIMZADEH, M.; ATTAR, A.; SAKHAEI, S. M. A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct scan dataset. **Biomedical Signal Processing and Control**, Elsevier, v. 68, p. 102588, 2021.

RAN, A. R.; CHEUNG, C. Y.; WANG, X.; CHEN, H.; LUO, L.-y.; CHAN, P. P.; WONG, M. O.; CHANG, R. T.; MANNIL, S. S.; YOUNG, A. L. *et al.* Detection of glaucomatous optic neuropathy with spectral-domain optical coherence tomography: a retrospective training and validation deep-learning analysis. **The Lancet Digital Health**, Elsevier, v. 1, n. 4, p. e172–e182, 2019.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 779–788.

RODRIGUES, L. S.; CAVALCANTI, I. M.; CAPANEMA, L. X. d. L.; MORCH, R. B.; MAGALHÃES, G.; LIMA, J. F.; BURNS, V. A. C.; JÚNIOR, A. J. A.; MUNGIOLI, R. P. Panorama da aquicultura no brasil: desafios e oportunidades. **BNDES Setorial**, n. 35, mar. 2012, p. 421-463, Banco Nacional de Desenvolvimento Econômico e Social, 2012.

RUDEK, M.; COELHO, L. d. S.; JR, O. C. Visão computacional aplicada a sistemas produtivos: Fundamentos e estudo de caso. **XXI Encontro Nacional de Engenharia de Produção-2001, Salvador**, 2001.

RUSSELL, S.; NORVIG, P. Artificial intelligence: a modern approach. 2002.

SALAZAR, R. D. V.; MESA, A. A. C.; OSORIO, L. Y. O. Propuesta de sistema de conteo de alevines de tilapia roja de bajo costo usando técnicas de visión artificial. **Revista Politécnica**, v. 11, n. 21, p. 85–95, 2015.

SENAR. **Piscicultura: manejo da produção de peixes em viveiros**. Brasília: Serviço Nacional de Aprendizagem Rural, 2017. ISBN 978-85-7664-178-0.

SENAR. **Piscicultura: reprodução, larvicultura e alevinagem de tilápias**. Brasília: Serviço Nacional de Aprendizagem Rural, 2017. ISBN 978-85-7664-171-1.

SIDDIQUE, F.; SAKIB, S.; SIDDIQUE, M. A. B. Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers. In: **2019 5th International Conference on Advances in Electrical Engineering (ICAEE)**. [S.l.: s.n.], 2019. p. 541–546.

SIMURRA, I.; KELLER, D.; FREITAS, B.; FERREIRA, J.; LEITE, C.; TEIXEIRA, M.; BESSA, W. R. B. A ferramenta soundsphere no contexto da exploração e da criação sonora: Um estudo de caso de música ubíqua dentro da sala de aula. In: **Anais do X Workshop de Música Ubíqua (UbiMus 2020)**. [S.l.: s.n.], 2020. v. 101, n. 104, p. 149.

SOUZA, D. C. de; NETO, F. M. M.; BAGA, O. C.; BESSA, W. R. B.; LEITE, D. G. Um sistema inteligente de monitoramento e avaliação da saúde do camarão. In: **ANAIS DO XXVII SEMIC UFERSA**. Mossóro: [s.n.], 2021.

SUTHERLAND, J.; VIKTOROV, A.; BLOUNT, J.; PUNTIKOV, N. Distributed scrum: Agile project management with outsourced development teams. In: IEEE. **2007 40th annual Hawaii international conference on system sciences (HICSS'07)**. [S.l.], 2007. p. 274a–274a.

TAN, P.; STEINBACH, M.; KUMAR, V. **Introdução ao datamining: mineração de dados**. Ciencia Moderna, 2009. ISBN 9788573937619. Disponível em: <<https://books.google.com.br/books?id=69d6PgAACAAJ>>.

TZUTALIN. **labelImg**. [S.l.]: GitHub, 2022. <<https://github.com/tzutalin/labelImg>>.

VALENCIA, I. J. C.; DADIOS, E. P.; FILLONE, A. M.; PUNO, J. C. V.; BALDOVINO, R. G.; BILLONES, R. K. C. Vision-based crowd counting and social distancing monitoring using tiny-yolov4 and deepsort. In: IEEE. **2021 IEEE International Smart Cities Conference (ISC2)**. [S.l.], 2021. p. 1–7.

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

VOGADO, L. H.; VERAS, R. M.; ARAUJO, F. H.; SILVA, R. R.; AIRES, K. R. Rede neural convolucional para o diagnóstico de leucemia. In: SBC. **Anais do XIX Simpósio Brasileiro de Computação Aplicada à Saúde**. [S.l.], 2019. p. 46–57.

WANG, C.-Y.; BOCHKOVSKIY, A.; LIAO, H.-Y. M. Scaled-YOLOv4: Scaling cross stage partial network. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2021. p. 13029–13038.

WONG, X. Y.; ANG, Y. K.; LI, K.; CHIN, Y. H.; LAM, S. S. W.; TAN, K. B. K.; CHUA, M. C. H.; ONG, M. E. H.; LIU, N.; POURGHADERI, A. R. *et al.* Development and validation of the sarica score to predict survival after return of spontaneous circulation in out of hospital cardiac arrest using an interpretable machine learning framework. **Resuscitation**, Elsevier, 2021.

WU, D.; LV, S.; JIANG, M.; SONG, H. Using channel pruning-based yolo v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. **Computers and Electronics in Agriculture**, Elsevier, v. 178, p. 105742, 2020.

XIMENES, L. F. Produção de pescado no brasil e no nordeste brasileiro. Banco do Nordeste do Brasil, 2021.

YANG, B.; GAO, Z.; GAO, Y.; ZHU, Y. Rapid detection and counting of wheat ears in the field using yolov4 with attention module. **Agronomy**, Multidisciplinary Digital Publishing Institute, v. 11, n. 6, p. 1202, 2021.

YU, J.; ZHANG, W. Face mask wearing detection algorithm based on improved yolo-v4. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 21, n. 9, p. 3263, 2021.



ZHAO, S.; ZHANG, S.; LU, J.; WANG, H.; FENG, Y.; SHI, C.; LI, D.; ZHAO, R. A lightweight dead fish detection method based on deformable convolution and yolov4. **Computers and Electronics in Agriculture**, Elsevier, v. 198, p. 107098, 2022.