

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica

Bruno Alberto Soares Oliveira

**SISTEMA DE DIAGNÓSTICO DA ESQUISTOSSOMOSE A
PARTIR DE IMAGENS MICROSCÓPICAS PREPARADAS
COM A TÉCNICA KATO-KATZ**

Belo Horizonte

2022

Bruno Alberto Soares Oliveira

**Sistema de Diagnóstico da Esquistossomose a Partir de Imagens
Microscópicas preparadas com a Técnica Kato-Katz**

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Doutor em Engenharia Elétrica.

Orientador: Frederico Gadelha Guimarães

Belo Horizonte

2022

O48s

Oliveira, Bruno Alberto Soares.

Sistema de diagnóstico da esquistossomose a partir de imagens microscópicas preparadas com a técnica Kato-Katz [recurso eletrônico] / Bruno Alberto Soares Oliveira. - 2022.

1 recurso online (150 f. : il., color.) : pdf.

Orientador: Frederico Gadelha Guimarães.

Tese (doutorado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f. 133-150.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Aprendizado do computador - Teses. 3. Aprendizado profundo - Teses. 4. Diagnóstico - Teses. 5. Esquistossomose - Teses. 6. Imagem médica - Teses. I. Guimarães, Frederico Gadelha. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
COLEGIADO DO CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FOLHA DE APROVAÇÃO

Sistema de Diagnóstico Automático da Esquistossomose a Partir de Imagens Microscópicas Preparadas com a Técnica Kato-Katz

Bruno Alberto Soares Oliveira

Tese de Doutorado defendida e aprovada, no dia **24 de março de 2022**, pela Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais constituída pelos seguintes professores:

Prof. Dr. Frederico Gadelha Guimarães - DEE (UFMG) - Orientador

Prof. Dr. Antônio de Pádua Braga - DELT (UFMG)

Prof. Dr. Pedro Pedrosa Rebouças Filho (IFCE)

Prof. Dr. João Paulo Papa - DCC (UNESP)

Prof. Dr. Deborah Aparecida Negrão-Corrêa - Parasitologia (UFMG)

Prof. Dr. Antonio Luiz Pinho Ribeiro - Clínica Médica (UFMG)

Belo Horizonte, 24 de março de 2022.



Documento assinado eletronicamente por **Frederico Gadelha Guimaraes, Professor do Magistério Superior**, em 16/04/2022, às 19:54, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

A autenticidade deste documento pode ser conferida no site



https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1273887** e o código CRC **168178CA**.

Referência: Processo nº 23072.210662/2022-61

SEI nº 1273887

Dedico este trabalho em primeiro lugar a Deus, que sempre me iluminou durante essa longa e dura jornada.

A minha mãe, que sempre orou por mim, torcendo pelo meu sucesso.

Ao meu pai, que sempre me incentivou e é um exemplo a ser seguido.

A minha noiva, que sempre esteve comigo no decorrer dessa caminhada.

Agradecimentos

Primeiramente, agradeço a Deus, que jamais saiu do meu lado. À minha mãe, que me ajudou a caminhar na fé e acreditar nos impossíveis da vida. Ao meu pai, exemplo de integridade, que me ensinou que a vida é um eterno sonhar e recomeçar. À minha noiva, que esteve comigo nas dificuldades e sucesso do curso. À minha madrinha, tias e tios, primos e primas, e a todos os meus familiares que sempre acreditaram em mim.

Agradeço, também, aos meus velhos amigos de infância, que até hoje caminham comigo. Também aos meus colegas que tiveram alegrias e tristezas, ao longo desses anos de curso. Aos meus professores, que sempre exigiram o máximo de mim e, ainda, a todos os funcionários da Universidade, que diretamente ou não, contribuíram para esse sucesso por mim alcançando.

A todos do laboratório MINDS, por viverem comigo essa experiência fantástica, enriquecendo toda pesquisa desenvolvida por mim durante essa temporada. Um agradecimento especial ao meu orientador, Prof. Frederico Gadelha Guimarães, por ter sido literalmente um orientador por todo esse tempo, parabenizando e criticando construtivamente todas as vezes em que se foi necessário. A equipe do Laboratório de Imunologia de Helminhos do Instituto de Ciências Biológicas, pela coleta e disponibilização dos dados que possibilitaram a pesquisa.

Por fim, agradeço a toda equipe do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais pela oportunidade de desenvolver o trabalho.

Obrigado! Vocês foram fundamentais nessa vitória!

*“Não importa o quão educado, talentoso, rico ou legal você acha que seja. O modo como
você trata as pessoas que dirá quem você realmente é!”*

(Autor desconhecido)

Resumo

Uma importante preocupação da área de saúde pública é causada pelos parasitos intestinais humanos, que são encontrados em grande parte nos países tropicais. O diagnóstico dessas doenças parasitárias se dá por meio de sintomas fisiológicos e exame fecal. Frequentemente, poucos profissionais estão disponíveis e aptos a realizarem esse tipo de exame, que é considerado lento, difícil, propenso a erros e pode causar fadiga ocular no especialista. Técnicas de inteligência artificial têm sido aplicadas com sucesso para problemas dessa natureza. Portanto, o objetivo desse trabalho é desenvolver uma solução baseada em aprendizado profundo e aprendizado de máquina para encontrar ovos de parasitos intestinais da espécie *Schistosoma mansoni*, sendo um sistema de auxílio a tomada de decisão no diagnóstico em exame de fezes cujas lâminas são preparadas por meio da técnica parasitológica Kato-Katz. Foi construída uma base de dados real com 1100 imagens que foram anotadas por três diferentes especialistas humanos no diagnóstico da esquistossomose. Técnicas de aumento de dados *online* e *offline* foram utilizadas para se obter um maior número de amostras e melhorar a generalização da ferramenta. Como resultado, a solução proposta atingiu um valor de AP de 0.884 para um @[IoU=0.50]. Os resultados e a empregabilidade do sistema são promissores, podendo vir a ser utilizado no SUS para auxiliar o profissional da saúde no diagnóstico da esquistossomose.

Palavras-chave: Aprendizado de Máquina, Aprendizado Profundo, Diagnóstico, Esquistossomose, Imagens Médicas, Kato-Katz.

Abstract

A major public health concern is caused by human intestinal parasites, which are found largely in tropical countries. The diagnosis of these parasitic diseases is made through physiological symptoms and fecal examination. Often, few professionals are available and able to perform this type of examination, which is considered slow, difficult, error-prone, and can cause eye fatigue in the specialist. Artificial intelligence techniques have been successfully applied to problems of this nature. Therefore, the objective of this work is to develop a solution based on deep learning and machine learning to find intestinal parasite eggs of the species *S. mansoni*, being a system to aid decision-making in the diagnosis of fecal examination whose slides were prepared using the Kato-Katz parasitological technique. A real database was built with 1100 images that were annotated by three different human specialists in the diagnosis of schistosomiasis. Data augmentation techniques *online* and *offline* were used to obtain a larger number of samples and improve the generalizability of the tool. As a result, the proposed solution achieved an AP value of 0.884 for an @[IoU=0.50]. The results and employability of the system are promising, and it could be used in the SUS to assist health professionals in diagnosing schistosomiasis.

Keywords: Deep Learning, Diagnosis, Kato-Katz, Machine Learning, Medical Imaging, Schistosomiasis

Lista de Ilustrações

Figura 1 – Exemplo de ovo da espécie <i>S. mansoni</i> em um campo microscópico com <i>zoom</i> de 40x (A), 100x (B), 200x (C) e 400x (D).	23
Figura 2 – Exemplo da utilização da técnica Kato-Katz em um campo microscópico.	24
Figura 3 – Esquema de um neurônio simples.	35
Figura 4 – Perceptron de múltiplas camadas.	36
Figura 5 – Esquema de uma rede neural convolucional.	39
Figura 6 – Exemplo da matriz de confusão.	41
Figura 7 – Interseção sobre união.	44
Figura 8 – Exemplos de interseção sobre união.	45
Figura 9 – Comparação entre classificação de imagens, detecção de objetos e segmentação de instâncias.	48
Figura 10 – Arquitetura do R-CNN.	52
Figura 11 – Exemplos de aplicação da pesquisa seletiva.	52
Figura 12 – Exemplo de funcionamento do algoritmo de pesquisa seletiva.	54
Figura 13 – Arquitetura da AlexNet.	54
Figura 14 – Arquitetura do <i>Fast</i> R-CNN.	56
Figura 15 – Exemplo da aplicação do <i>pool</i> de RoI.	57
Figura 16 – Arquitetura do <i>Faster</i> R-CNN.	58
Figura 17 – Funcionamento da RPN.	59
Figura 18 – Comparativo entre a velocidade do tempo de inferência para cada um dos métodos de detecção de objetos baseados em regiões.	60
Figura 19 – Modelo do método YOLO.	61
Figura 20 – Saída do método YOLO.	62
Figura 21 – Arquitetura do método YOLO.	63
Figura 22 – Pipeline do método YOLO.	63

Figura 23 – Arquitetura da estrutura Darknet-19.	64
Figura 24 – Previsão da caixa delimitadora.	66
Figura 25 – Arquitetura da estrutura Darknet-53.	67
Figura 26 – Arquitetura do método SSD.	68
Figura 27 – Funcionamento da previsão de caixas delimitadoras do método SSD.	69
Figura 28 – Imagens de <i>S. mansoni</i> disponibilizadas pela Universidade Kansas State.	75
Figura 29 – Diagrama de blocos ilustrando as etapas de desenvolvimento do estudo.	81
Figura 30 – Exemplo de uso do LabelImg pelo especialista.	83
Figura 31 – Exemplos de imagens em que as anotações dos especialistas divergiram.	84
Figura 32 – Distribuição das áreas das caixas delimitadoras do especialista A.	85
Figura 33 – Distribuição das áreas das caixas delimitadoras do especialista B.	85
Figura 34 – Boxplot das áreas das caixas delimitadoras dos especialistas A e B.	86
Figura 35 – Aplicação do aumento de dados <i>offline</i>	89
Figura 36 – Diagrama do <i>pipeline</i> de processo.	92
Figura 37 – Estratégia <i>k-fold</i> utilizada para divisão do conjunto de dados.	96
Figura 38 – Procedimento para criação do conjunto de imagens utilizado para o treinamento e teste dos métodos de aprendizado de máquina.	97
Figura 39 – Exemplos da aplicação da técnica HOG em imagens da classe “schisto”.	99
Figura 40 – Exemplos da aplicação da técnica HOG em imagens da classe “não schisto”.	100
Figura 41 – Sistema proposto.	105
Figura 42 – Gráfico da variação dos limiares de detecção para o modelo <i>Faster</i> <i>R-CNN</i> com uma arquitetura de CNN ResNet-50.	110
Figura 43 – Gráfico normalizado da variação dos limiares de detecção para o modelo <i>Faster R-CNN</i> com uma arquitetura de CNN ResNet-50.	111
Figura 44 – Matriz de confusão do modelo KNN.	112
Figura 45 – Matriz de confusão do modelo <i>Naive Bayes</i>	112
Figura 46 – Matriz de confusão do modelo <i>Random Forest</i>	113
Figura 47 – Matriz de confusão do modelo SVM.	114
Figura 48 – Matriz de confusão do modelo <i>XGBoost</i>	114
Figura 49 – Exemplo de casos em que a integração entre o detector de objetos de aprendizado profundo e os modelos de aprendizado de máquina atrapalhou no diagnóstico final da esquistossomose.	119

Figura 50 – Anotações realizadas pelos especialistas A, B e C nas duas imagens em que os modelos de aprendizado de máquina classificaram incorretamente dois ovos da espécie <i>S. mansoni</i>	120
Figura 51 – Exemplos de detecção em que a integração do método <i>Faster</i> R-CNN com os modelos de aprendizado de máquina ajudaram a amenizar as detecções incorretas.	121
Figura 52 – Exemplos de imagens sem ovos de <i>S. mansoni</i> e que a integração do método <i>Faster</i> R-CNN com os modelos de aprendizado de máquina ajudaram a amenizar a quantidade de falsos positivos.	122
Figura 53 – Detecções realizadas pela solução apresentada nesse estudo em imagens utilizadas nos trabalhos relacionados.	124
Figura 54 – Imagens detectadas pela ferramenta ParasiteID.	125
Figura 55 – Tela inicial do sistema proposto.	126
Figura 56 – Tela inicial após o usuário selecionar uma imagem.	127
Figura 57 – Tela da aplicação que apresenta o resultado de um caso em que havia ovos da espécie <i>S. Mansonii</i> na imagem.	128
Figura 58 – Tela da aplicação que apresenta o resultado de um caso em que não foi detectado nenhum ovo da espécie <i>S. Mansonii</i> na imagem.	128

Lista de Tabelas

Tabela 1 – Redes Neurais Convolucionais para Visão Computacional.	49
Tabela 2 – Exemplo hipotético da saída gerada por um detector de objetos.	50
Tabela 3 – Resumo dos trabalhos relacionados.	78
Tabela 4 – Métricas de avaliação para detecção de objetos em COCO.	94
Tabela 5 – Preparação dos dados para os modelos de aprendizado de máquina. . .	101
Tabela 6 – Parâmetros de treinamento dos modelos de aprendizado de máquina testados.	102
Tabela 7 – Resultados obtidos para o comparativo entre as anotações dos especialistas.	106
Tabela 8 – Resultados obtidos para os experimentos com diferentes métodos de detecção de objetos.	107
Tabela 9 – Variação dos limiares de detecção para o modelo <i>Faster</i> R-CNN com uma arquitetura de CNN ResNet-50.	109
Tabela 10 – Resultados obtidos pelos modelos de aprendizado de máquina treinados para a tarefa de classificação.	115
Tabela 11 – Variação dos limiares de detecção para o modelo <i>Faster</i> R-CNN inte- grado aos classificadores de aprendizado de máquina.	117
Tabela 12 – Resultados obtidos com as imagens utilizadas nos trabalhos relacionadas.	123

Lista de Abreviaturas e Siglas

AP	<i>Average Precision</i>
CNN	Rede Neural Convolutacional
FN	Falso Negativo
FP	Falso Positivo
HOG	<i>Histogram of Oriented Gradients</i>
IA	Inteligência Artificial
IoU	Intersecção sobre União
KNN	<i>K - Nearest Neighbors</i>
MINDS	Machine Intelligence and Data Science
MLP	Perceptron Multicamadas
MCSVM	Máquina de Vetores de Suporte Multiclasse
RNA	Rede Neural Artificial
SUS	Sistema Único de Saúde
SVM	Máquina de Vetores de Suporte
UFMG	Universidade Federal de Minas Gerais
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

Sumário

1	Introdução	14
1.1	Motivação	14
1.2	Objetivo Geral	18
1.3	Objetivos Específicos	18
1.4	Contribuições da Tese	18
1.5	Organização do Trabalho	21
2	Referencial Teórico	22
2.1	Problema de Interesse	22
2.2	Método Kato-Katz	23
2.3	Aprendizado de Máquina	25
2.3.1	Aprendizado Supervisionado	26
2.3.2	Problema de Classificação	27
2.3.3	<i>K - Nearest Neighbors</i>	27
2.3.4	<i>Naive Bayes</i>	28
2.3.5	<i>Random Forest</i>	30
2.3.6	<i>Support Vector Machine</i>	31
2.3.7	<i>XGBoost</i>	32
2.4	Aprendizado Profundo	33
2.4.1	Aprendizado por Transferência	34
2.5	Redes Neurais Artificiais	35
2.6	Redes Neurais Convolucionais	37
2.7	Aumento de Dados	40
2.8	Métricas de Avaliação dos Resultados	40
2.8.1	Matriz de Confusão	41
2.8.2	Métricas de Qualidade	42
2.8.2.1	Acurácia	42
2.8.2.2	Precisão (especificidade)	43
2.8.2.3	Revocação (sensibilidade)	43
2.8.2.4	F1-score	43

2.8.3	Precisão Média e Interseção sobre União	44
2.9	Resumo	45
3	Detecção de Objetos	47
3.1	Tarefas de Visão Computacional	47
3.2	R-CNN	51
3.2.1	Fast R-CNN	55
3.2.2	Faster R-CNN	58
3.3	YOLO	60
3.3.1	YOLOv2	63
3.3.2	YOLOv3	65
3.4	SSD	66
3.5	Resumo	70
4	Trabalhos Relacionados	72
4.1	Pesquisas	72
4.2	Resumo	79
5	Material e Métodos	81
5.1	Coleta e pré-processamento do conjunto de imagens	82
5.2	Definição dos métodos de detecção de objetos	90
5.3	Experimentos e análise dos resultados	91
5.3.1	Comparativo com imagens utilizadas nos trabalhos relacionados	95
5.4	Integração com modelos de aprendizado de máquina	96
5.5	Comparativo com aplicativo comercial	102
5.6	Sistema para produção	103
5.7	Resumo	104
6	Resultados e Discussão	105
6.1	Comparativo entre as anotações dos especialistas	106
6.2	Métodos de detecção de objetos	107
6.3	Seleção do limiar de detecção para o modelo de detecção de objetos	109
6.4	Métodos de aprendizado de máquina	111
6.5	Integração entre o modelo de detecção de objetos e os modelos de aprendi- zado de máquina	116
6.6	Comparativo com imagens dos trabalhos relacionados	123
6.7	Comparativo com a ferramenta ParasiteID	125

6.8 API	126
6.9 Resumo	129
7 Conclusões	130
Referências	133

Capítulo 1

Introdução

1.1 Motivação

Estima-se atualmente que mais de um bilhão e meio de pessoas encontram-se infectadas por ao menos uma espécie de helminto no mundo, onde a maior parte destes indivíduos residem em áreas subdesenvolvidas da África, Ásia e Américas (WHO, 2021; Naidoo et al., 2021). Dentre as helmintoses de maior impacto na saúde humana destaca-se a esquistossomose, que se caracteriza por ser uma doença crônica e debilitante causada pela infecção por parasitos do gênero *Schistosoma*. Atualmente, estão descritas cinco espécies deste trematódeo capazes de infectar seres humanos, sendo elas: *S. hematobium*, *S. intercalatum*, *S. japonicum*, *S. mansoni* e *S. mekongi* (Chuah et al., 2019). Estima-se que a doença esquistossomose afeta cerca de 220 milhões de pessoas que vivem em áreas tropicais e subtropicais de 78 países nos continentes da África, América e Ásia (Gryseels et al., 2006; Colley et al., 2014; WHO, 2020).

No Brasil, a esquistossomose humana é causada exclusivamente pela espécie *S. mansoni*, estimando-se que 1,5 milhões de brasileiros estejam infectados pelo parasito (Katz, 2018). As principais áreas endêmicas brasileiras estão localizadas no litoral Nordeste e nos estados da Bahia, Minas Gerais e Espírito Santo, enquanto nos estados do Rio de Janeiro e São Paulo, em áreas das regiões Norte e Sul do país a transmissão desta parasitose possui caráter focal e disperso (Colley et al., 2014; Zoni et al., 2016; Carvalho et al., 2018). Inicialmente os estados da Bahia e Minas Gerais caracterizavam-se por altas prevalências de infecção e indivíduos que possuíam moderadas/altas cargas parasitárias. Atualmente, estas e outras áreas endêmicas têm sofrido grandes modificações no perfil de

transmissão da esquistossomose em consequência de medidas de melhorias de estrutura sanitária, atendimento médico em áreas endêmicas e educação em saúde.

Outro fator primordial para essa mudança se deve à atuação do Programa de Controle da Esquistossomose (PCE) realizado em áreas endêmicas, cujos principais pilares são o diagnóstico e o tratamento dos indivíduos infectados. Esta estratégia de controle tem sido associada à redução da carga parasitária nos indivíduos positivos para *S. mansoni*, ocasionando a redução dos casos mais graves de hepatoesplenomegalia, todavia não possui efeito protetor contra reinfecções, permitindo a expansão da esquistossomose para outras regiões do país (Savioli et al., 2017; Coelho et al., 2016; Ross et al., 2015; Katz, 2018; Nascimento et al., 2019; Silva-Moraes et al., 2019). Entretanto, mesmo com os esforços das medidas de controle da doença no Brasil, a esquistossomose mansoni ainda é responsável por uma taxa de mortalidade de 0,4 para cada 100 mil habitantes, e a morbidade associada à esquistossomose no país foi estimada em 102,259 dias de vidas perdidas por incapacidade ou morte prematura (DALYs) por 100 mil habitantes (95% UI 59,76-176,12), sendo relatada como a segunda causa de morbidade entre as doenças negligenciadas no ano de 2016 (Katz, 2018; Martins-Melo et al., 2018).

A técnica diagnóstica mais amplamente utilizada para detectar infecções por *S. mansoni* é a técnica de Kato-Katz, por apresentar alta especificidade, baixo custo, facilidade operacional e principalmente por possibilitar a análise de uma maior quantidade de material fecal por lâmina, em comparação com outras técnicas parasitológicas. Este exame coproscópico detecta ovos do parasito nas fezes sendo recomendado pela Organização Mundial da Saúde (OMS) para o diagnóstico laboratorial da esquistossomose mansoni (Ash et al., 1994). A técnica de Kato-Katz é eficiente na detecção de infecções moderadas e intensas; entretanto, indivíduos com baixas cargas parasitárias não são facilmente identificadas por este método (De Vlas and Gryseels, 1992; Oliveira et al., 2018). Além disso, esta técnica possui baixa reprodutibilidade, quando se examina diferentes lâminas do mesmo indivíduo; leituras díspares são obtidas por examinadores distintos e apresenta resultados de falso negativos, quando aplicada em amostras de fezes de pacientes que estão em período pré-patente (Kongs et al., 2001; Berhe et al., 2004).

Para contornar estes problemas, técnicas que se utilizam de abordagens distintas têm sido testadas, como ensaios moleculares que visam identificar sequências genômicas do parasito em amostras de material biológico como sangue e fezes. Apesar de apresentarem

bons desempenhos diagnósticos, testes moleculares requerem um maior recurso financeiro e infraestrutura mais sofisticada para sua realização (Pontes et al., 2002; Espírito-Santo et al., 2014; Weerakoon et al., 2018; Fuss et al., 2018). Ensaios imunológicos que visam à detecção de anticorpos específicos contra *S. mansoni* têm se mostrado eficazes. A técnica mais comumente utilizada é o ensaio de ELISA, entretanto, como limitação, já foi demonstrada a possibilidade de a resposta humoral permanecer elevada por um longo período após a cura parasitológica e a ocorrência de reações cruzadas com outros helmintos (Rabello et al., 2008; Cavalcanti et al., 2013; Hinz et al., 2017; Fuss et al., 2018).

Dessa forma, por mais que tenham surgido alternativas, ainda assim a técnica de Kato-Katz é a mais empregada para essa situação. Especialistas treinados, a partir de seu conhecimento prévio e com o auxílio de um microscópio, examinam o material fecal do paciente em busca de ovos de parasitos. Salienta-se que todo esse processo é realizado de forma manual pelo especialista, em que é comum acontecer erros de diagnóstico devido ao cansaço, fadiga e falta de experiência do profissional (De Carli, 2001; Neves, 2003), resultando em altas taxas de falsos negativos, especialmente em casos em que há um baixo número de ovos no material fecal (Tarafder et al., 2010). No trabalho de Barbosa et al. (2017), a qualidade das lâminas preparadas pela técnica Kato-Katz para o diagnóstico da esquistossomose foi avaliada. Os testes foram realizados com dois especialistas treinados em que a taxa de concordância entre eles foi de 70%, enquanto a discordância 30%. O estudo discute sobre o quanto é difícil encontrar ovos da espécie *S. mansoni* em imagens microscópicas preparadas pela técnica Kato-Katz, até mesmo para profissionais treinados.

Para amenizar o problema da falta de especialistas treinados para uma tomada de decisão correta e diminuir o tempo necessário para o diagnóstico a partir do exame parasitológico manual de fezes, é necessário o desenvolvimento de tecnologias que possam ser capazes de automatizar e auxiliar nesse processo. A detecção automática de ovos de parasitos em exames fecais permitiria a inspeção de um maior número de amostras com um alto grau de confiabilidade e objetividade. Tecnologias atuando nesses problemas seriam úteis, principalmente, nos países onde se tem um alto índice de pessoas infectadas por meio de doenças parasitológicas.

Nos últimos anos os estudos em aprendizado profundo vêm se difundindo e suas aplicações vêm se tornando cada vez mais presentes no dia a dia da sociedade. Aprendizado profundo pode ser entendido como uma família de métodos de aprendizado de máquina

que são baseados em redes neurais artificiais (RNA). Esse tipo de aprendizado pode ser supervisionado, semi-supervisionado ou não supervisionado.

Atualmente, um dos principais algoritmos de aprendizado de máquina na área de visão computacional são as redes neurais convolucionais, também conhecidas como CNNs ou ConvNets. Elas pertencem a uma classe de redes neurais profundas e são constantemente aplicadas em problemas de classificação e detecção de objetos em imagens (Goodfellow et al., 2016).

Uma rede neural convolucional é um algoritmo de aprendizado profundo que recebe uma imagem como entrada. Logo depois, essa imagem é processada e o algoritmo extrai as características mais relevantes de algum objeto pré-determinado contido na imagem para que, em seguida, consiga diferenciar um ou mais objetos baseados nos padrões aprendidos. Feito isso, o algoritmo consegue rotular cada um dos objetos e dizer a qual classe pertencem (LeCun et al., 2015).

Dessa forma, a hipótese de pesquisa a ser investigada nessa Tese de Doutorado é se a integração de técnicas de aprendizado de máquina clássicas e métodos de detecção de objetos baseados em aprendizado profundo podem contribuir no diagnóstico da esquistossomose. Sendo assim, as principais contribuições dessa pesquisa são:

- Coleta de imagens microscópicas com lâminas preparadas pela técnica Kato-Katz contendo ovos da espécie *S. mansoni*.
- Desenvolvimento de uma metodologia de integração de técnicas de aprendizado de máquina clássicas e métodos de detecção de objetos baseados em aprendizado profundo para encontrar e contar ovos de parasitos intestinais humanos da espécie *S. mansoni* em imagens microscópicas com lâminas preparadas pela técnica Kato-Katz.
- Solução automatizada para apoio a tomada de decisão dos profissionais da área médica no problema abordado.
- Sistema viável e funcional para ser colocado em produção e utilizado na prática por especialistas.

1.2 Objetivo Geral

O objetivo deste estudo foi desenvolver um sistema automático de auxílio ao especialista na tomada de decisão para o diagnóstico da esquistossomose. A solução foi baseada em métodos de detecção de objetos eficientes para encontrar e contar ovos de parasitos intestinais humanos da espécie *S. mansoni* em imagens microscópicas com lâminas preparadas pela técnica Kato-Katz.

1.3 Objetivos Específicos

Um dos objetivos específicos e essencial para o desenvolvimento da pesquisa foi o de construção do conjunto de imagens, uma vez que não foi encontrada uma base de dados pública com imagens de ovos de parasitos da espécie *S. mansoni* em imagens microscópicas preparadas pela técnica parasitológica Kato-Katz. Por isso, foi necessário construir e preparar o conjunto de dados por conta própria.

Além disso, fez parte do escopo do trabalho avaliar o quão eficiente poderia ser integrar técnicas de aprendizado de máquina clássicas e métodos de detecção de objetos baseados em aprendizado profundo para o diagnóstico da esquistossomose. Para isso, foi necessário implementar, treinar, avaliar e testar os modelos abordados.

Com os experimentos, analisar e discutir os resultados obtidos para que fosse possível definir um modelo de inteligência artificial para fazer parte do sistema automático de auxílio ao especialista na tomada de decisão para o diagnóstico da esquistossomose. Por fim, desenvolver uma ferramenta de análise de imagens que possa interagir com o especialista ao ser colocada em produção.

1.4 Contribuições da Tese

Durante o desenvolvimento desse trabalho de doutorado foram publicados artigos que se relacionam com o tema desse estudo, os quais influenciaram diretamente no produto final gerado. Além disso, várias outras pesquisas também foram desenvolvidas que se correlacionam com as áreas de Inteligência e Visão Computacional. Os principais trabalhos relacionados com a área da parasitologia foram:

- *Automated Diagnosis of Schistosomiasis by using Faster R-CNN for Egg Detection in Microscopy Images Prepared by the Kato-Katz Technique (Neural Computing and Applications - 2022)*: no artigo foi verificada a capacidade do método de detecção de objetos *Faster R-CNN* em encontrar ovos de parasitos da espécie *S. mansoni* em imagens microscópicas. O conjunto de imagens era bem menor em relação ao utilizado na tese de doutorado, o que limitou os resultados obtidos.
- Uma Abordagem de Classificação de Ovos de Parasitos Intestinais Humanos em Imagens Microscópicas usando Redes Neurais Convolucionais (Congresso Brasileiro de Inteligência Computacional - 2021): na pesquisa foi proposta uma arquitetura de CNN específica para o problema de classificação de ovos de parasitos de cinco diferentes espécies. Com o trabalho foi possível personalizar a topologia da rede, servindo como base para o estudo realizado na tese de doutorado.
- *Deep Learning Convolutional Networks for Automating Recognition of Human Intestinal Parasites in Microscopy Images (Expert Systems With Applications - revisões requeridas)*: no trabalho foi feito um comparativo entre diferentes arquiteturas de CNN para a classificação de ovos de cinco espécies de ovos de parasitos. A tese aqui apresentada é uma evolução do artigo, que não se limita apenas à classificação, mas também à detecção dos ovos.
- Solução apresentada na tese de doutorado: existe a possibilidade do registro de *software* do sistema proposto junto ao Instituto Nacional da Propriedade Industrial (INPI).

Os trabalhos desenvolvidos com o tema de Inteligência e Visão Computacional foram:

- *Automated Construction Management Platform with Image Analysis Using Deep Learning Neural Networks (Machine Learning with Applications - em revisão)*: o artigo apresenta um comparativo entre diferentes métodos de detecção de objetos para o reconhecimento de elementos que compõe alimentadores em subestações de energia elétrica. O estudo é similar ao da tese, mas em um contexto diferente.
- *Automated monitoring of construction sites of electric power substations using deep learning (IEEE Access - 2021)*: no trabalho foi investigada a capacidade de diferentes

arquiteturas de CNN para o reconhecimento de elementos que compõe alimentadores em subestações de energia elétrica. Com o estudo foi possível conhecer e explorar tais arquiteturas, que também foram utilizadas na tese de doutorado.

- *Explaining Machine Learning based Diagnosis of COVID-19 from Routine Blood Tests with Decision Trees and Criteria Graphs (Computers in Biology and Medicine - 2021)*: no estudo foi desenvolvida uma técnica para diagnóstico da COVID-19. Foram explorados vários classificadores de aprendizado de máquina, que também serviram como base para o uso deles na tese.
- *Visão Computacional e Aprendizado de Máquina para Classificação de Componentes em Alimentadores de Subestações Elétricas (Congresso Brasileiro de Automática - 2020)*: no artigo foram avaliadas diferentes arquiteturas de CNN para a classificação de elementos que compõe alimentadores em subestações de energia elétrica. O uso das arquiteturas possibilitou conhecimento para serem empregados aqui na tese.
- *Predição de Risco de Doença Cardiovascular Como Estratégia de Prevenção Primária Utilizando Aprendizado de Máquina e Modelos Explicáveis (Congresso Brasileiro de Engenharia Biomédica -2020)*: vários foram os métodos de aprendizado de máquina abordados no trabalho, possibilitando o entendimento desses modelos e os empregando na tese de doutorado.
- *Deep Learning para Detecção de Componentes em Alimentadores de Subestações (Simpósio Brasileiro de Telecomunicações e Processamento de Sinais - 2020)*: na pesquisa foi utilizado o método de detecção de objetos YOLO para a detecção de componentes em alimentadores de subestações de energia elétrica. O mesmo método também foi empregado aqui na tese.
- *Rastreamento Ocular como Ferramenta para Melhorar o Diagnóstico do Espectro do Autismo (Congresso Brasileiro de Engenharia Biomédica - 2020)*: foram abordados diferentes métodos de aprendizado de máquina para melhorar o diagnóstico do espectro do autismo. As técnicas empregadas também foram utilizadas aqui na tese de doutorado.
- *Sistema para Classificação de Grãos de Café: Uma Comparação entre Redes Neurais Convolucionais e Processamento Digital de Imagens com MLP (Simpósio Brasileiro*

de Automação Inteligente - 2019): na pesquisa foi realizado um comparativo entre uma arquitetura específica de CNN e técnicas clássicas de extração de recursos utilizando processamento digital de imagens. O trabalho serviu como auxílio ao uso das técnicas de seleção de características assim como as utilizadas na tese.

1.5 Organização do Trabalho

Essa tese é organizada da seguinte forma:

No capítulo 2 é apresentado o referencial teórico, em que são definidos importantes conceitos para o bom entendimento do trabalho, como o problema de interesse, o que é a técnica parasitológica Kato-Katz, aprendizado de máquina, *deep learning*, redes neurais artificiais, redes neurais convolucionais, as técnicas de *data augmentation* e sobre as métricas de avaliação dos resultados utilizadas nessa pesquisa.

O capítulo 3 apresenta superficialmente as principais tarefas na área da visão computacional, além de conceituar e explicar o funcionamento dos métodos de detecção de objetos que compõem o estado da arte nesse tipo de tarefa utilizando aprendizado profundo. Já o capítulo 4 apresenta e discute os resultados dos estudos de outros autores utilizando técnicas de inteligência artificial na detecção de ovos de parasitos.

No capítulo 5 são apresentados os materiais e os métodos utilizados no desenvolvimento desse trabalho, como a criação da base de imagens utilizada, como foi definida a arquitetura da CNN proposta e como o modelo foi avaliado. Aqui são descritas todas as etapas de aquisição do material e realização dos experimentos, de forma que o trabalho possa ser executado por outros pesquisadores, para a replicabilidade.

O capítulo 6 apresenta os resultados obtidos após os experimentos realizados, os quais foram descritos no capítulo 5. Neste capítulo se tem a discussão dos resultados encontrados para as arquiteturas propostas após a realização dos experimentos variando algumas configurações de treinamento. Já o capítulo 7 fornece as considerações conclusivas, algumas limitações e indica oportunidades de novas pesquisas.

Capítulo 2

Referencial Teórico

No capítulo de Referencial Teórico são apresentados os mais importantes conceitos, justificativas e características sobre o assunto abordado. É nesse capítulo que são descritas e explicadas em detalhes grande parte das técnicas adotadas no desenvolvimento do trabalho, bem como o problema pesquisado.

2.1 Problema de Interesse

Atualmente, as técnicas realizadas no setor de parasitologia são exclusivamente manuais, podendo ser influenciadas possivelmente por variáveis não controláveis como a atenção e a experiência do técnico laboratorial. A identificação dos ovos de parasitos é determinada microscopicamente, por um profissional habilitado, de acordo com a morfologia dos ovos. Em especial para o caso da espécie *S. mansoni*, os pacientes possuem uma baixa carga parasitária de ovos em relação a outras espécies de parasitos, dificultando ainda mais o diagnóstico da esquistossomose. Além disso, os altos investimentos do setor público e privado no PCE fortalecem a justificativa da escolha do *S. mansoni* nessa pesquisa.

A Figura 1 (A) ilustra um campo microscópico de esfregaço fecal de Kato-Katz obtido em lâminas com poucas fezes e com *zoom* de 40x, aumento que foi utilizado no conjunto de imagens deste estudo. A Figura 1 (B) possui um *zoom* de 100x e em (C) 200x. A Figura 1 (D) apresenta uma imagem de ovo da espécie *S. mansoni* com um *zoom* de 400x, no qual é possível observar todos os detalhes morfológicos dos ovos desta espécie. Uma linha branca foi desenhada em cada uma das imagens, que tem 10mm de tamanho, apenas para exemplo.

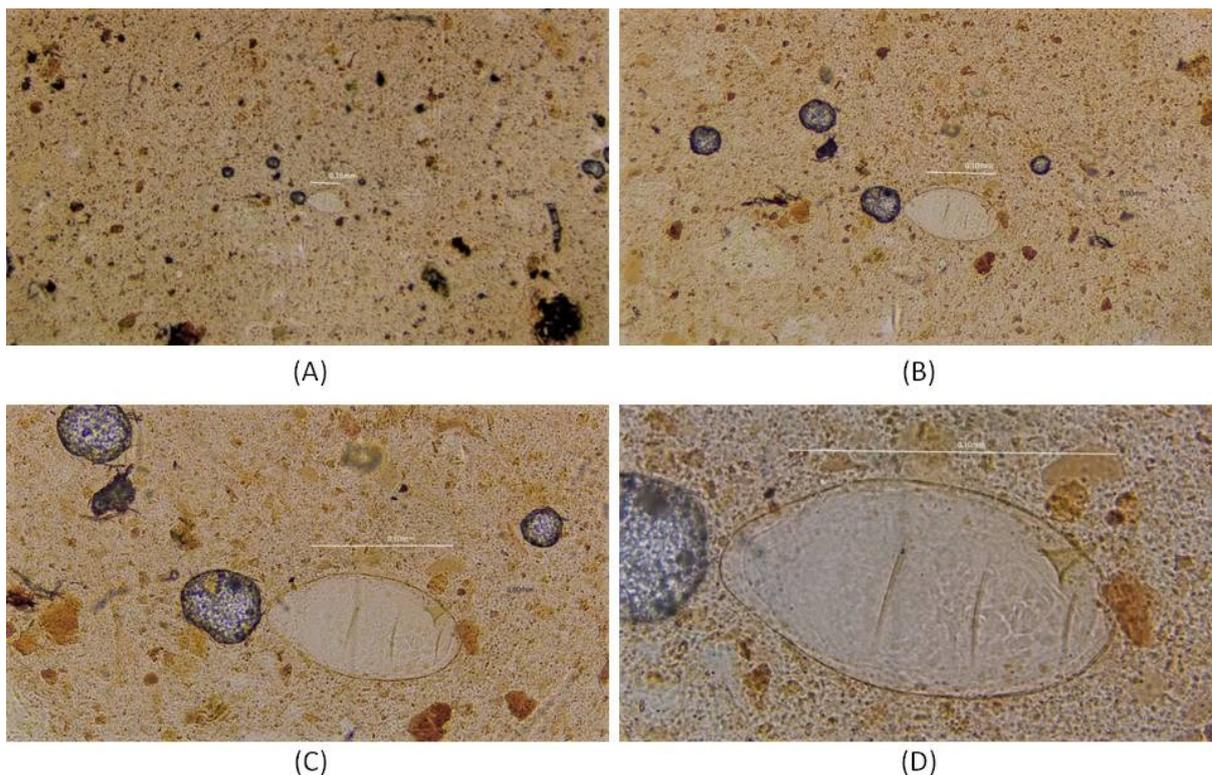


Figura 1 – Exemplo de ovo da espécie *S. mansoni* em um campo microscópico com *zoom* de 40x (A), 100x (B), 200x (C) e 400x (D).

Como pode ser observada na Figura 1, a tarefa de detecção de ovos de parasitos em uma amostra fecal não é intuitiva para ser feita manualmente. A maior dificuldade que os especialistas enfrentam é o alto acúmulo de sujeira, fungos, bolhas de água, entre outras coisas na placa, o que dificulta a localização dos ovos. Um outro problema é a quantidade elevada de falsos negativos que existem nos diagnósticos com esse tipo de análise, que é quando o especialista deixa de notar algum ovo na amostra fecal e dá ao paciente um diagnóstico equivocado, o que não é incomum em casos de baixa carga parasitária (Enk et al., 2008; Siqueira et al., 2015; Barbosa et al., 2017; Oliveira et al., 2018).

2.2 Método Kato-Katz

O método Kato-Katz é uma técnica quantitativa empregada quando se deseja conhecer a carga parasitária em uma lâmina, ou seja, estimar a quantidade de ovos de helmintos que um determinado indivíduo está eliminando em suas fezes. É possível utilizar essa técnica a partir de fezes frescas ou conservadas em formaldeído. Em fezes preservadas, elas não podem estar no estado líquido e o conservante empregado deve ser descartado no

momento do exame (Katz et al., 1972).

Devido a sua simplicidade e objetividade, o método Kato-Katz é a principal técnica de diagnóstico de presença de helmintos e atualmente é considerada a técnica mais utilizada em exames de rotina nos serviços públicos e privados de saúde e em laboratórios de pesquisa, devido à sua rápida preparação e à facilidade em se obter as amostras para varredura.

A Figura 2 ilustra um campo microscópico (ampliação de 100x) de um esfregaço fecal de Kato-Katz, contendo ovos de Ancilostomídeo (A), *Ascaris lumbricoides* (B), *Enterobius vermicularis* (C) e *Trichuris trichiura* (D), cada um marcado pela cor vermelha, que foi previamente detectado por um especialista.

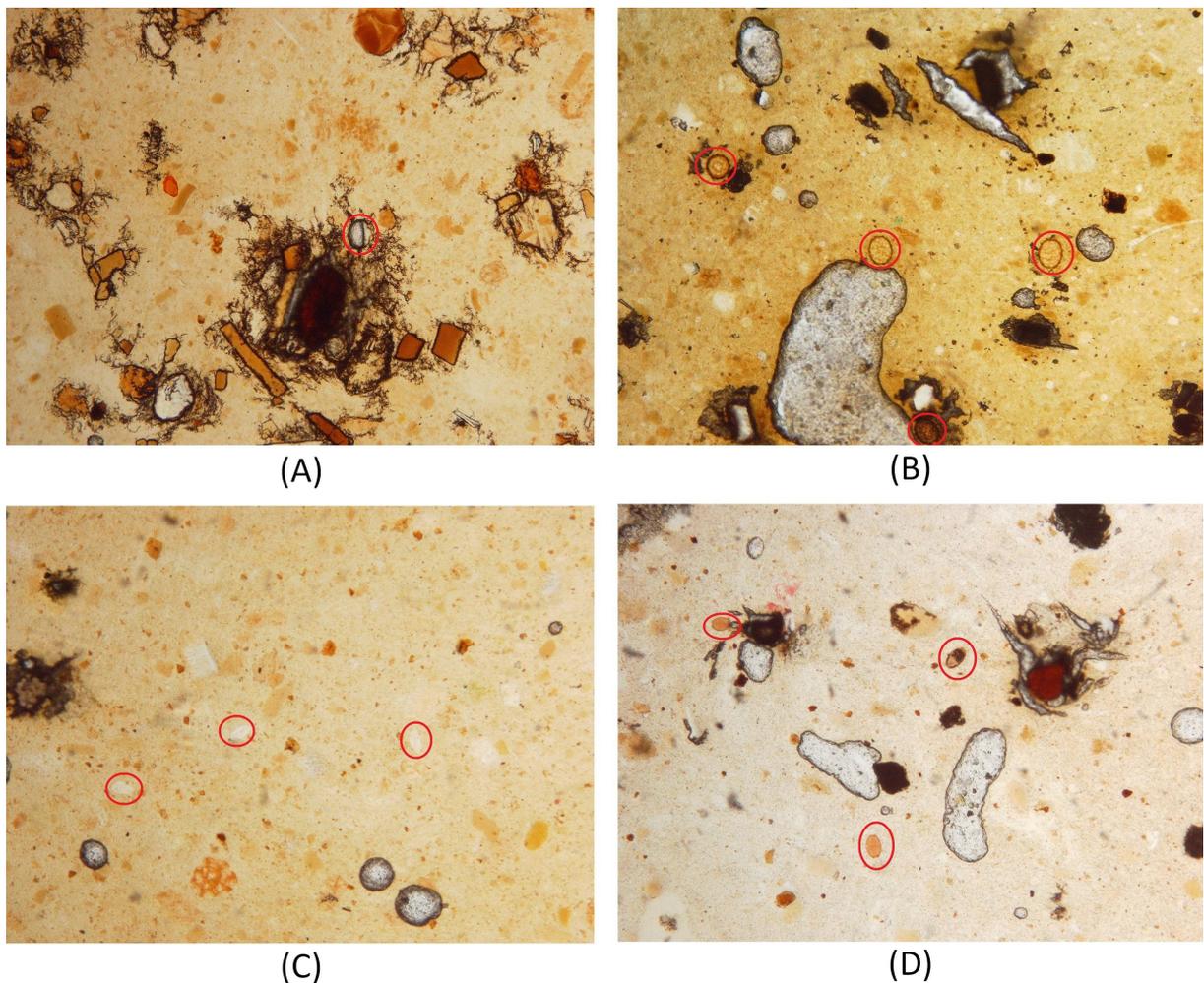


Figura 2 – Exemplo da utilização da técnica Kato-Katz em um campo microscópico.

Os materiais necessários para uso dessa técnica são: o kit Kato-Katz, um microscópio binocular, algumas máscaras, avental, luvas e lâminas. Para preparação das lâminas é

necessário que a amostra seja realizada a partir de fezes frescas (ou mantidas na geladeira por até 48 horas) e não diarreicas. Com os materiais disponíveis, o técnico laboratorial irá seguir uma determinada metodologia padrão conforme descrita por [Barbosa et al. \(2017\)](#). Posteriormente a realização desse procedimento e com o auxílio de um microscópio óptico, o técnico irá conseguir de forma mais clara, visualizar os ovos de helmintos contidos na amostra fecal.

Existem algumas limitações do método, sendo que em grande parte das dificuldades dos técnicos laboratoriais se baseiam na ausência de ovos nas fezes. Mesmo para aqueles indivíduos que possuem alguma doença parasitária pode acontecer de, devido a certas circunstâncias, tais indivíduos não possuírem uma alta carga parasitária, levando geralmente a casos de diagnósticos que possuem como resultado o falso negativo ([Barbosa et al., 2017](#)).

2.3 Aprendizado de Máquina

Para [Michie et al. \(1994\)](#), aprendizado de máquina é uma aplicação de inteligência artificial (IA) que fornece aos sistemas computacionais uma certa capacidade de aprender e melhorar de forma autônoma a partir da sua própria experiência, fazendo com que não se torne necessário que tal sistema seja explicitamente programado. O aprendizado de máquina se concentra no desenvolvimento de programas de computador que podem acessar dados e usá-los para aprender por si próprios.

O processo de aprendizado começa com observações ou dados, como exemplos, a fim de procurar padrões nos dados e tomar melhores decisões no futuro com base nos exemplos que foram fornecidos. O principal objetivo é permitir que os computadores aprendam automaticamente sem que haja intervenção ou assistência humana e ajustem seus parâmetros implicitamente buscando resolver problemas de classificação ou de regressão.

Os algoritmos de aprendizado de máquina permitem a análise de grandes quantidades de dados. Embora geralmente ofereça resultados mais rápidos e precisos para identificar oportunidades lucrativas ou riscos perigosos, também pode exigir tempo e recursos adicionais para realizar a etapa de treinamento do modelo adequadamente. A combinação de aprendizado de máquina com IA e tecnologias cognitivas pode tornar essa classe de algoritmos ainda mais eficiente no processamento de grande volume de

informações (Castoril et al., 2018; Nunes et al., 2020).

2.3.1 Aprendizado Supervisionado

Os algoritmos de aprendizado de máquina são geralmente classificados como supervisionados ou não supervisionados. O aprendizado supervisionado é um tipo de sistema no qual são fornecidos os dados de entrada e de saída desejados. Os dados de entrada e saída são rotulados para fornecer uma base de aprendizado com o intuito de classificar futuramente dados não rotulados (Rosenberg et al., 2005).

O aprendizado supervisionado ocorre quando um sistema recebe variáveis de entrada e saída com a intenção de aprender como tais variáveis são mapeadas. O objetivo é descobrir uma função de mapeamento para que, quando novas entradas desconhecidas forem fornecidas, o modelo possa prever a saída. Esse é um processo iterativo e, toda vez que o algoritmo faz uma previsão, ele é corrigido ou recebe um *feedback* até atingir um nível aceitável de desempenho (Rosenberg et al., 2005). Em outras palavras, o aprendizado supervisionado resolve f na Equação 2.1.

$$Y = f(X) \tag{2.1}$$

Um dos métodos de atualização dos parâmetros internos bastante utilizado no treinamento de redes multicamada é o *backpropagation*, proposto por Hecht-Nielsen (1992). De forma resumida, o método funciona inicialmente calculando o erro da camada de saída da arquitetura do modelo, comparando com o valor esperado. Os pesos entre a camada de saída e a camada anterior são atualizados para diminuir o erro e assim o gradiente do erro é repassado à camada anterior. Esse processo é realizado até que todos os pesos até a camada de entrada sejam atualizados.

Os dados de treinamento para essa classe de algoritmos incluem um conjunto de exemplos com características de entrada relacionados e uma saída desejada. Por exemplo, em uma aplicação de aprendizado supervisionado com imagens, um sistema de IA pode receber imagens rotuladas de veículos em categorias como carros ou caminhões. Após o processo de observação dessas imagens e a etapa de treinamento concluída, o sistema deve ser capaz de distinguir e categorizar imagens não identificadas, como pertencente a uma das duas classes.

As aplicações do aprendizado supervisionado são tipicamente divididas em duas categorias, classificação e regressão. A classificação é semelhante ao exemplo acima, quando o valor de saída é uma categoria, como um carro ou um caminhão. Um problema de regressão ocorre quando a saída é um valor real calculado, como preço, peso, temperatura ou umidade.

2.3.2 Problema de Classificação

No campo do aprendizado de máquina, um problema de classificação consiste em identificar corretamente a classe à qual uma determinada amostra pertence, com base em um conjunto de dados de treinamento que contém observações de mesma categoria conhecida. Alguns exemplos amplamente utilizados que buscam explicar esse tipo de problema são os casos de classificação de *e-mails*, em que os mesmos podem ser classificados como spam ou como não spam (Alpaydin, 2014).

Um algoritmo que implementa a classificação, especialmente em uma implementação concreta, é conhecido como classificador. O termo classificador às vezes também se refere a uma função matemática, implementada por um algoritmo de classificação que mapeia os dados de entrada para uma determinada categoria. São vários os métodos de aprendizado de máquina supervisionados para resolver problemas de classificação atualmente, sendo que cada um possui suas especificidades.

2.3.3 *K - Nearest Neighbors*

O classificador *K - Nearest Neighbors* (KNN) é de longe um dos algoritmos mais simples de aprendizado de máquina utilizado para classificação de imagens. O método foi desenvolvido no trabalho proposto por Fix and Hodges (1989) e aperfeiçoado no estudo de Altman (1992).

Internamente, o KNN depende da distância entre os vetores de recursos extraídos das imagens. Cada uma das imagens está associada a uma classe ou rótulo, permitindo que o algoritmo avalie esses recursos no espaço para posteriormente medir as distâncias entre tais características de cada imagem.

De uma forma simples, o método KNN classifica os pontos de dados desconhecidos encontrando seus vizinhos mais próximos, conforme ressaltado em seu próprio nome. Sendo

assim, quanto mais vizinhos de uma determinada classe tem uma imagem desconhecida, a mesma pertencerá justamente a essa classe majoritária entre seus vizinhos. O método KNN parte da crença de que imagens semelhantes possuem distâncias menores umas às outras, em comparação com imagens que possuem características muito diferentes.

Para se aplicar o método KNN é necessário a definição de dois parâmetros importantes: uma métrica de distância e um valor de “K”. Para a distância, as mais comuns são a distância Euclidiana e a distância de Manhattan. Outras métricas de distância também podem ser utilizadas dependendo da natureza dos dados. Já o valor do “K” do KNN é o número de vizinhos que será abordado para cada uma das amostras que será classificada. A Equação 2.2 apresenta a equação da distância euclidiana e a Equação 2.3 apresenta a equação da distância de Manhattan.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.2)$$

$$d(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2.3)$$

2.3.4 *Naive Bayes*

O algoritmo *Naive Bayes* é um dos métodos de aprendizado de máquina para tarefas de classificação (Rish et al., 2001), sendo considerado de fácil entendimento e implementação. Além disso, é um algoritmo de aprendizagem supervisionada, fazendo parte da classe dos classificadores probabilísticos.

O *Naive Bayes* é um algoritmo que consegue rotular uma amostra como sendo pertencente a classe A ou a classe B. Também, faz parte dos métodos de aprendizado supervisionado, os quais aprendem examinando exemplos previamente identificados. Primeiro, o algoritmo aprende a partir de um conjunto de exemplos do treino para que, em seguida, consiga classificar amostras contidas no subconjunto de teste que não haviam participado durante a etapa de treinamento do modelo.

Naive Bayes também é um classificador probabilístico. A classe que o algoritmo determina para uma nova instância é o resultado da criação da distribuição de probabilidade de todas as classes aprendidas pelo modelo e, em seguida, toma a decisão de escolher

a classe que obteve uma maior probabilidade. Algoritmos de classificação probabilística avaliam a distribuição de probabilidade condicional, ou seja, a probabilidade de atribuir uma classe a um conjunto específico de características de entrada (Lowd and Domingos, 2005).

O algoritmo faz uso do teorema de Bayes como princípio fundamental. A Equação 2.4 mostra a definição formal do Teorema de Bayes para calcular um evento (A), considerando que um evento (B) ocorreu.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2.4)$$

onde,

- $P(A|B)$: Probabilidade do evento A acontecer.
- $P(B|A)$: Probabilidade de B acontecer, dado que A já ocorreu.
- $P(A)$: Probabilidade de A ocorrer.
- $P(B)$: Probabilidade de B acontecer.

Dessa forma, com base no Teorema de Bayes, o classificador *Naive Bayes* pode ser definido como uma função que atribui um rótulo de classe $\hat{y} = C_k$ para alguns k conforme descrito na Equação 2.5.

$$\hat{y} = \arg \max_{k \in \{1, \dots, K\}} p(C_k) \prod_{i=1}^n p(x_i | C_k) \quad (2.5)$$

Consequentemente, o método de *Naive Bayes* possui como principais vantagens:

- Simplicidade.
- Requer um conjunto pequeno para treinamento.
- É um método computacionalmente rápido.
- Consegue lidar com classificação binária e multiclasse.

As principais desvantagens são:

- Supõe que os recursos das instâncias são independentes (o que raramente acontece em aplicações reais).
- Por causa da suposição anterior, pode fornecer estimativas ruins.

2.3.5 *Random Forest*

Em português, Floresta Aleatórias, é um algoritmo de aprendizado supervisionado, podendo ser utilizado tanto em tarefas de classificação quanto regressão. O método foi proposto no trabalho de [Breiman \(2001\)](#) e é considerado um dos algoritmos de aprendizado de máquina mais empregados para a solução de problemas em diversas áreas do conhecimento ([Goldstein et al., 2010](#); [Belgiu and Drăguț, 2016](#); [Naghbi et al., 2017](#)).

Nos conceitos da natureza, sabe-se que uma floresta é composta por várias árvores e, quanto mais árvores existam naquele determinado local, mais robusta é a floresta. Dito isso, o algoritmo atua com essa mesma premissa, criando árvores de decisão ([Breiman et al., 2017](#)) em amostras de dados selecionadas aleatoriamente, obtendo resultados de classificação em cada uma das árvores e por fim, selecionando a classe da nova instância que foi eleita majoritariamente.

O método das Floresta Aleatórias funciona basicamente por meio de quatro etapas:

1. Selecionam-se amostras aleatórias em um determinado conjunto de dados.
2. Constrói-se uma árvore de decisão para cada amostra e se obtém um resultado de classificação para cada uma das árvores de decisão geradas.
3. Executa-se uma votação para cada um dos resultados previstos.
4. Por meio de votação, é selecionada a classe que foi mais repetida em cada árvore de decisão gerada, encontrando o resultado final.

Algumas vantagens que podem ser exaltadas do algoritmo Floresta Aleatórias são:

- As Floresta Aleatórias são consideradas um método altamente preciso e robusto devido ao número de árvores de decisão que participam do processo.
- Geralmente não sofre um problema constantemente enfrentado por algoritmos de aprendizado de máquina, o *overfitting*. A principal razão que faz com que as Floresta

Aleatórias não possuem *overfitting* está no fato de que o método considera todas as previsões realizadas, diminuindo os vieses.

- O algoritmo pode ser utilizado em problemas de classificação e regressão.
- As Floresta Aleatórias lidam bem com valores ausentes. Existem duas formas para contornar este problema: usar valores medianos para substituir as variáveis contínuas e computar a média ponderada de acordo com a proximidade dos valores faltantes.
- É possível obter as características mais relevantes das amostras, permitindo assim que o cientista de dados selecione apenas as características que mais contribuem para o modelo.

No mais, a técnica também possui como principal desvantagem:

- Tem um custo computacional alto para gerar as classificações justamente porque possui várias árvores de decisão. Sempre que uma árvore toma uma decisão para um valor de entrada, este mesmo valor também é submetido às demais árvores da floresta, elevando seu custo computacional.

Em resumo, o método Floresta Aleatórias é um conjunto de várias árvores de decisão, com a vantagem de sofrer menos *overfitting* que uma árvore trabalhando sozinha. Contudo, uma única árvore de decisão tem a vantagem de ser computacionalmente mais rápida e ser mais fácil de explicar as decisões tomadas, uma vez que ela é facilmente interpretável quando convertida em regras de decisão.

2.3.6 *Support Vector Machine*

Em português, Máquina de Vetores de Suporte (SVM), é um método de aprendizado de máquina supervisionado que pode ser empregado tanto em tarefas de classificação quanto em tarefas de regressão. Contudo, é principalmente utilizado em problemas de classificação. No algoritmo SVM, cada instância é avaliada como um ponto no espaço n -dimensional (onde n é a quantidade de características da amostra) como cada recurso sendo o valor de uma coordenada específica no plano.

A técnica foi proposta inicialmente no trabalho de [Boser et al. \(1992\)](#) e aperfeiçoada no estudo de [Cristianini et al. \(2000\)](#). Atualmente o algoritmo SVM é um dos métodos

mais utilizados e poderosos dentre as técnicas de aprendizado de máquina. Até o advento e a crescente fama do *deep learning*, em português, aprendizado profundo, ele era o mais utilizado em tarefas de classificação mais complexas, tendo sido ultrapassado pelas redes neurais artificiais, objeto de estudo das próximas seções.

De maneira geral, uma boa curva de separação entre as classes é conseguida pelo hiperplano que possui uma maior distância entre as amostras de treinamento, cada uma pertencente a sua classe. Dessa forma, quanto maior a distância da curva de separação entre os pontos da classe, menor é o erro do classificador. Contudo, em problemas práticos, são raras as vezes que as classes são linearmente separáveis no espaço pertencente. Por esta razão, o método SVM consegue propor que os dados sejam mapeados em uma dimensão muito superior, fazendo com que a curva de separação entre as classes seja mais bem traçada no espaço do problema. Dessa forma, o algoritmo consegue ser eficiente e aplicado em diversas áreas do conhecimento desde o início do século até os dias atuais (Byvatov and Schneider, 2003; Noble et al., 2004; Deka et al., 2014; Zendejboudi et al., 2018; Al-Thanoon et al., 2019; Syriopoulos et al., 2021).

2.3.7 *XGBoost*

O classificador *XGBoost* é um método de aprendizado de máquina aplicado a dados estruturados e tabulares. O algoritmo é baseado na abordagem de árvores de decisão com aumento do gradiente, o que faz ele ser mais rápido e possuir um melhor desempenho. O fato do *XGBoost* empregar a técnica de aumento de gradiente extremo (Chen et al., 2015), faz com que ele possua muitos módulos. Sendo assim, ele é considerado um algoritmo *ensemble* (Chen and Guestrin, 2016).

Uma abordagem *ensemble* é um conjunto de algoritmos que tomam uma decisão compartilhada, isto é, a classificação resultante é dada com base em um entre todos os métodos considerados no conjunto (Dietterich, 2000). Sendo assim, a abordagem *ensemble* pode empregar diferentes métodos de aprendizado de máquina ou até mesmo apenas um, com a obrigatoriedade de ser no mínimo dois ou possuir mais métodos, mesmo que se tratando da mesma técnica.

Para que o classificador *XGBoost* consiga utilizar a abordagem *ensemble*, várias árvores de decisão são consideradas. Por meio da técnica *boosting*, o algoritmo é treinado

seguindo uma cadeia sequencial para o aprendizado. Cada uma das árvores de decisão são treinadas sequencialmente e, caso a árvore atual realize uma classificação incorreta, o erro existente é retificado, passando-o para a próxima árvore, e assim sucessivamente. Dessa forma, o algoritmo *XGBoost* consegue ser robusto e aplicável em diversos problemas da sociedade (Zhang and Zhan, 2017; Pan, 2018; Lim and Chi, 2019; Parsa et al., 2020)

2.4 Aprendizado Profundo

Aprendizado profundo (do inglês, *Deep learning*), é uma técnica de aprendizado de máquina que visa “ensinar” as máquinas a agirem e interpretarem dados de uma maneira mais natural (Deng et al., 2014). O aprendizado profundo tem diversas aplicações, como por exemplo em veículos autônomos, em que o automóvel consegue reconhecer e identificar o estado atual de um semáforo, para que tome a decisão de parar antes do cruzamento ou prosseguir pela via.

Atualmente, o aprendizado profundo vem sendo aplicado em diversos ramos da pesquisa e da indústria, pelo fato de que essa técnica tem alcançado resultados que antes não eram possíveis. Até então, métodos tradicionais realizavam o trabalho de extração de características de forma manual, ficando a cargo do desenvolvedor definir as melhores estratégias. Com o advento do aprendizado profundo, essa seleção dos melhores recursos é realizada de forma automática, implicando diretamente nos resultados obtidos.

Utilizando a técnica de aprendizado profundo, um modelo é capaz de aprender a executar corretamente a tarefa de classificação de imagens, texto ou som. Tais modelos podem alcançar resultados extremamente precisos e, em muitos casos, possuem desempenho superior ao de um humano. Os modelos são treinados usando um grande conjunto de dados rotulados e arquiteturas de redes neurais artificiais que possuem muitas camadas ocultas.

De acordo com Schmidhuber (2015), para se conseguir resultados precisos e satisfatórios ao se utilizar aprendizado profundo, basicamente é necessário entender duas premissas. A primeira é que lidar com o aprendizado profundo requer uma grande quantidade de dados rotulados, ou seja, é necessário milhares de imagens de cada espécie de ovo de parasito, para se ter um modelo que classifica corretamente tais espécies por meio de imagens, considerando a morfologia de cada ovo. A segunda é que aprendizado

profundo requer um alto poder computacional, tal que seja necessário a utilização de GPUs, configuração de arquitetura paralela, entre outros.

Grande parte dos métodos de aprendizado profundo utilizam arquiteturas de redes neurais artificiais e, por isso, tais modelos são frequentemente chamados de redes neurais profundas. Geralmente as redes neurais tradicionais possuem apenas duas ou três camadas ocultas, enquanto as redes neurais profundas podem ter centenas. Tal profundidade permite que o modelo aprenda características diretamente dos dados rotulados, sem a necessidade de uma extração manual de características.

2.4.1 Aprendizado por Transferência

Atualmente existem três maneiras de se criar e treinar modelos de aprendizado profundo que têm como objetivo realizar a classificação de objetos, são eles: treinamento a partir do zero, extração de recursos e o *transfer learning*, em português, aprendizado por transferência. No treinamento a partir do zero é definido uma nova arquitetura da rede e seus parâmetros são inicializados aleatoriamente e, dependendo da quantidade de dados e a taxa de aprendizado, esse tipo de treinamento pode levar dias ou até semanas para que se conclua o aprendizado. A abordagem da extração de recursos usa a rede como tal extrator, em que se aprende recursos a partir de imagens e posteriormente utilizá-los como um modelo de aprendizado de máquina, como uma *Support Vector Machine* (SVM), por exemplo (Erhan et al., 2010).

Para Pan and Yang (2009), grande parte dos modelos de aprendizado profundo desenvolvidos atualmente utilizam essa abordagem de aprendizado por transferência, o qual é caracterizado por ser um processo que envolve o ajuste fino dos parâmetros em um modelo que já foi previamente treinado, utilizando outros dados e outros recursos.

O processo de aprendizado por transferência se dá início com uma rede já existente, como a AlexNet (Krizhevsky et al., 2012) ou GoogleNet (Szegedy et al., 2015). Posteriormente, com os dados rotulados, treina-se um classificador com os exemplos do novo problema. É possível, na etapa de treinamento, ajustar alguns parâmetros da rede pré-treinada como por exemplo, ajustar os pesos sinápticos de apenas algumas camadas da arquitetura, congelando as demais. A maior vantagem desse tipo de abordagem se dá pelo fato de que é necessário um número bem menor de dados, de forma que o tempo de

computação na etapa de treinamento reduza consideravelmente.

O processo de aprendizado por transferência requer uma configuração adequada de interface com os parâmetros da rede pré-treinada, para que tais valores possam ser ajustados e aprimorados para a nova tarefa. Atualmente existem inúmeras bibliotecas que auxiliam o desenvolvedor que opta por utilizar esse tipo de abordagem em seus modelos.

2.5 Redes Neurais Artificiais

As primeiras implementações das RNAs tiveram como inspiração os sistemas neurais biológicos, porém, atualmente, esses dois sistemas possuem poucas semelhanças. No começo dos estudos sobre o tema, foi desenvolvido uma estrutura computacional que continha semelhanças a um simples neurônio natural, que recebia dados de entrada, multiplicava esses dados por valores reais chamados pesos sinápticos, combinava o resultado dessas multiplicações produzindo um valor de ativação, que era passado como parâmetro à outra função para gerar um dado de saída, como pode ser visto na Figura 3. Com o avanço dos estudos, os então chamados neurônios artificiais começaram a ser combinados em diferentes topologias de redes, criando-se, então, as RNAs (Barreto, 2002).

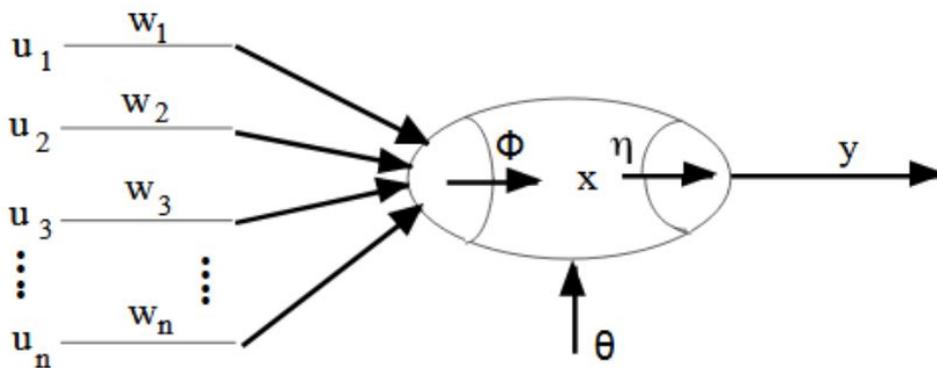


Figura 3 – Esquema de um neurônio simples.

Fonte: Adaptado de Hagan et al. (1997).

Uma definição para uma RNA seria um sistema dinâmico complexo, pois através das ligações entre neurônios ela se torna uma rede de sistemas interconectados, que pode ser representada por um grafo ponderado e direcionado, onde os vértices são as conexões entre os neurônios, os pesos dos vértices são os pesos sinápticos e os nós do grafo representam

os componentes do neurônio que fazem as combinações dos valores e geram as saídas (Barreto, 2002).

Um modelo perceptron simples é composto por um único neurônio e consegue realizar apenas separações lineares cujos objetos podem ser classificados em apenas dois grupos. Com a combinação de vários perceptrons, em uma topologia com camadas de neurônios, foi possível classificar objetos não-linearmente separáveis em muitos grupos, como também realizar aproximações de funções complexas e previsões de dados (Flores, 2009).

A Figura 4 ilustra uma rede perceptron multicamadas (MLP) com x_n entradas de dados em sua camada de entrada, que são multiplicados pelos pesos sinápticos e combinados em n_1 neurônios na primeira camada escondida. As saídas de cada camada oculta são multiplicadas por um novo conjunto de pesos, alimentando a próxima camada e sendo por fim direcionadas a m neurônios na camada de saída. Após aplicada uma função de ativação, são gerados y_m valores de probabilidade, como saída da rede.

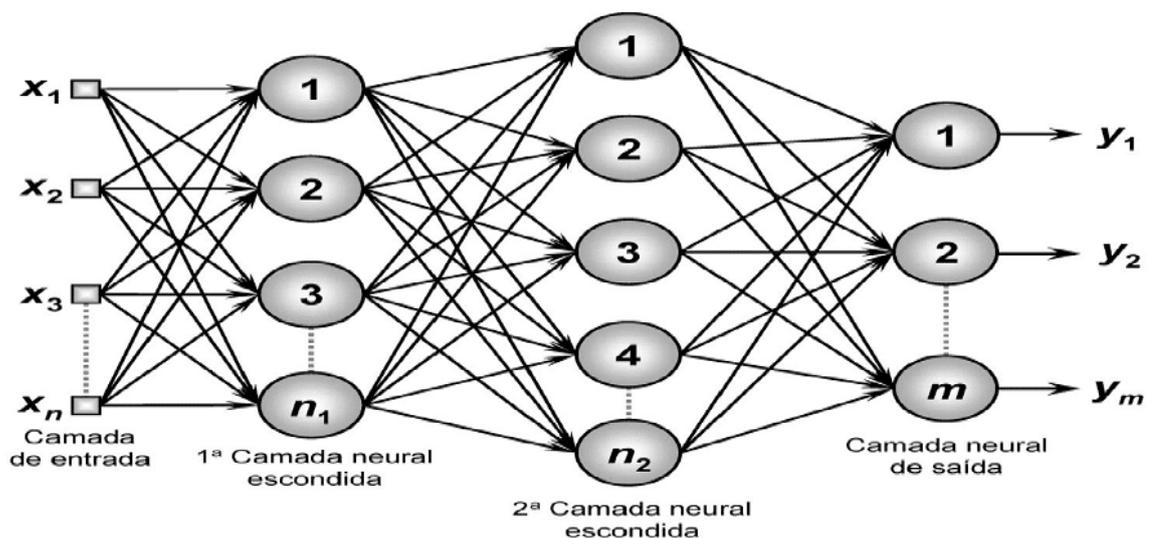


Figura 4 – Perceptron de múltiplas camadas.

Fonte: Adaptado de Hagan et al. (1997).

Dada uma saída durante a etapa de treinamento, é necessário que a rede avalie se a saída está correta ou não. Para isso, é calculada a diferença entre a saída real e a prevista pela rede. Como o objetivo é reduzir o erro da predição, o valor calculado é utilizado para a atualização dos pesos da rede, por meio do algoritmo *backpropagation*, que calcula o gradiente da função de erro em relação aos pesos atuais da rede.

2.6 Redes Neurais Convolucionais

A literatura considera que os principais conceitos das redes neurais convolucionais se deram pelo trabalho de [Hubel and Wiesel \(1962\)](#), que mostrou que os neurônios de alguns animais respondem individualmente a pequenas regiões do campo visual. Foi esse estudo que engajou vários outros trabalhos futuros até chegarmos recentemente na era do aprendizado profundo.

Em 1995, no trabalho de [LeCun et al. \(1995\)](#), os autores utilizaram esse estudo para arquitetar a primeira rede neural convolucional, em inglês *convolutional neural network* (CNN). O nome dado é justificado pelo fato de que a rede emprega uma operação matemática chamada convolução, que é um tipo especial de operação linear. CNNs são simplesmente redes neurais que usam a operação de convolução ([Bengio et al., 2017](#)).

De acordo com [Szegedy et al. \(2015\)](#) e [LeCun et al. \(2010\)](#), as CNNs são redes do tipo *feedforward* e tipicamente possuem as seguintes camadas em sua arquitetura: camadas convolucionais, camadas de subamostragem (ou *Pooling*) e camadas densas ou completamente conectadas, que podem ser dispostas de diversas formas a depender do problema abordado.

As camadas de convolução são responsáveis por extrair as características mais significativas da imagem. O processo de convolução é dado através da aplicação de extratores de características na imagem original. Uma das maiores vantagens da CNN em relação aos demais métodos de processamento digital de imagens é que nessa abordagem é possível obter os parâmetros dos extratores de características durante a etapa de treinamento da CNN, isto é, não se faz necessário a participação de um especialista para a escolha das características (atributos) de treinamento do modelo de classificação. A Equação 2.6 apresenta o cálculo das operações em uma camada convolucional. Considera-se que cada camada l possui I^l canais de entrada e O^l canais de saída, onde $q = 1, \dots, O^l$, sendo que os canais de saída gerados são os novos mapas de características.

$$C_q^l = \sigma \left(\sum_{p=1}^{N_{input}^l} C_p^{l-1} * k_{p,q}^l + b_q^l \right) \quad (2.6)$$

Conforme demonstrado no trabalho de [Zhang \(2016\)](#), um termo de viés (b^l) também é somado na camada de convolução em cada um dos mapas de características.

A última operação da camada convolucional é uma função de ativação σ que insere uma não linearidade aos dados. O operador $*$ representa a operação de convolução, que é apresentada em detalhes na Equação 2.7, onde, r e c representam a largura e comprimento das máscaras de convolução.

$$C_q^l(i, j) = \sigma \left(\sum_{p=1}^{N_{input}^l} \sum_{u=-r/2}^{r/2} \sum_{v=-c/2}^{c/2} C_p^{l-1}(i-u, j-v) \cdot k_{p,q}^l + b_q^l \right) \quad (2.7)$$

As camadas de subamostragem, também conhecidas como *Pooling*, segmentam as entradas das camadas anteriores em agrupamentos menores, reduzindo a sensibilidade da saída às pequenas variações e nuances da entrada de dados. Geralmente se aplica uma função de *pool* máximo nos dados de entrada da camada de subamostragem, uma vez que se espera obter as características mais relevantes da imagem. A Equação 2.8 mostra como é obtido o tamanho do mapa de características após a operação de *pool*, onde W^1 é a largura da matriz de entrada, H^1 é altura dessa mesma matriz e D^1 a profundidade da estrutura de entrada.

$$\begin{aligned} W^2 &= \frac{(W^1 - F)}{S + 1} \\ H^2 &= \frac{(H^1 - F)}{S + 1} \\ D^2 &= D^1 \end{aligned} \quad (2.8)$$

Para a aplicação da Equação 2.8 dado um volume de entrada, dois parâmetros são exigidos: o tamanho do kernel representado pela letra F e o passo S . Com isso, uma nova matriz W^2 , H^2 , D^2 (largura, altura e profundidade resultante) é gerada.

No fim da arquitetura de uma CNN existe a camada de achatamento, que é responsável por redimensionar uma imagem 2D, em um vetor 1D, que posteriormente será entrada para algum classificador, seja uma MLP, uma SVM, ou qualquer outro método que possua esse propósito. A Figura 5 ilustra um esquema de uma arquitetura de rede neural convolucional.

O treinamento de uma CNN é similar a outras técnicas mais simples de redes neurais artificiais (Bishop, 2006), uma vez que existe uma minimização da função de perda usando gradiente descendente. Desenvolver uma nova arquitetura de CNN para um

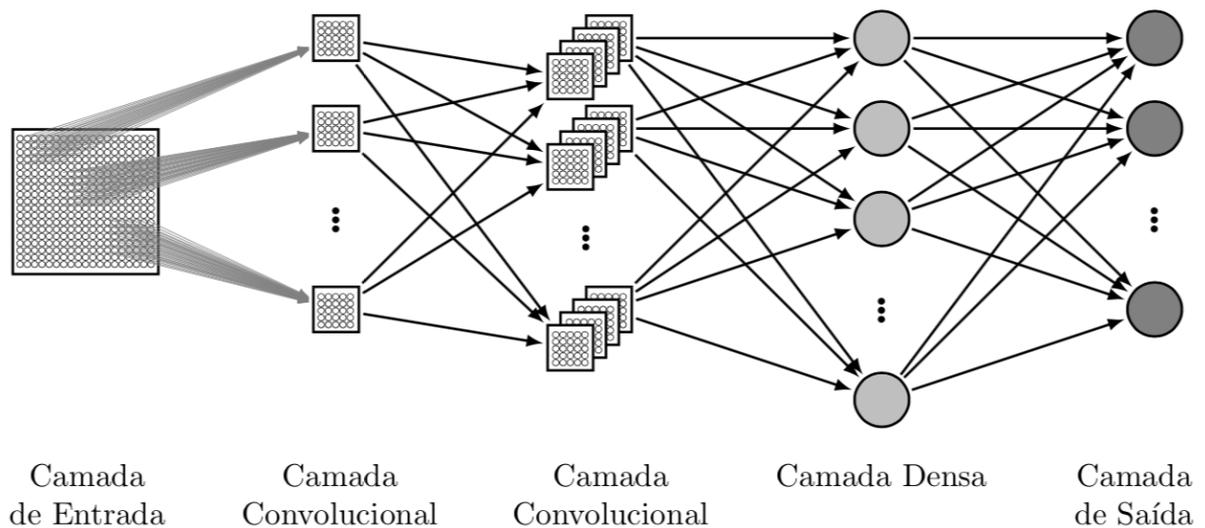


Figura 5 – Esquema de uma rede neural convolucional.

Fonte: Adaptado de [Xu et al. \(2014\)](#).

problema específico não é trivial dada a dependência da configuração de cada camada e dos vários parâmetros do problema a ser resolvido.

A eficiência das CNNs no processo de reconhecimento de imagens é uma das principais razões pelas quais o mundo acordou para a eficácia da aprendizagem profunda. Atualmente, existem inúmeros centros de pesquisas que estão impulsionando grandes avanços na visão computacional, que tem aplicações bastante úteis para carros autônomos, robótica, drones, segurança, diagnósticos médicos e tratamentos para deficientes visuais.

Existem algumas arquiteturas de CNN consolidadas e pré-treinadas que são aplicadas a problemas de reconhecimento de imagens por meio do aprendizado de transferência, como a MobileNet ([Howard et al., 2017](#)), a Xception ([Chollet, 2017](#)), DenseNet ([Huang et al., 2017a](#)), entre outras. As arquiteturas citadas pré-treinadas são utilizadas para classificar com sucesso diferentes tipos de objetos, como pessoas, carros, aviões, animais, entre outros. Porém, nenhum desses objetos possui uma aparência semelhante aos ovos do parasito *Schistosoma mansoni*. Por isso, um dos objetivos do estudo foi encontrar uma arquitetura de CNN juntamente com um método de detecção de objetos que fossem adequados para o reconhecimento dos ovos.

2.7 Aumento de Dados

Conforme dito por [Goodfellow et al. \(2016\)](#), técnicas de aprendizado profundo requerem milhares de dados para que seus modelos possam obter desempenhos satisfatórios. A escassez e o desbalanceamento de dados é um problema recorrente na área de aprendizado profundo, haja vista que uma grande parte dos problemas a serem resolvidos são extremamente difíceis ou até mesmo pode nem existir uma quantidade suficiente de dados para que o modelo a ser implementado possa ter o resultado esperado. Por isso, faz-se o uso de técnicas de aumento de dados.

O aumento de dados abrange uma ampla gama de técnicas usadas para gerar novas amostras de treinamento a partir das originais, aplicando agitações e perturbações aleatórias (mas ao mesmo tempo garantindo que os rótulos de classe dos dados não sejam alterados). O principal objetivo ao aplicar o aumento de dados é aumentar a generalização do modelo.

No contexto da visão computacional, o aumento de dados é altamente recomendável, pois é possível obter novos dados a partir das imagens originais aplicando transformações geométricas simples, como: translações, rotações, mudanças na escala, cisalhamento, inversões horizontais, entre outras. A aplicação de transformações em uma imagem de entrada mudará sua aparência levemente, mas não altera o rótulo da classe, tornando a ampliação de dados um método muito natural e fácil de ser aplicado para tarefas de aprendizado de máquina que incidem em imagens ([Perez and Wang, 2017](#)).

2.8 Métricas de Avaliação dos Resultados

Durante o processo de criação de um modelo de aprendizado de máquina é necessário que se meça a qualidade do mesmo em relação a sua eficiência em resolver determinada tarefa. Existem funções matemáticas que são qualificadas para avaliar a capacidade de erro e acerto de tal modelo, também chamadas de métricas.

Assim como é de extrema importância escolher um bom modelo para resolver tal problema, também é necessário escolher uma métrica que seja capaz de avaliar o quão bom foi o desempenho do modelo proposto.

Existem dezenas de métricas disponíveis nessa etapa de avaliação ([Sokolova et al.](#),

2006; Koyejo et al., 2014; Carvalho et al., 2019), sendo algumas simples e outras mais complexas. Algumas métricas podem funcionar melhor para determinados conjuntos de dados devido a suas características e peculiaridades.

Ao escolher uma métrica deve-se levar em consideração fatores como a proporção de dados de cada classe no conjunto de dados e o objetivo da classificação ou previsão (probabilidade, binário, *ranking*, entre outros). Por isso é importante conhecer bem a métrica que será utilizada, pois isso pode fazer toda a diferença no momento da avaliação do modelo. Nenhuma das métricas pode ser considerada melhor do que as outras em todos os casos. É sempre importante levar em conta a aplicação prática do modelo.

2.8.1 Matriz de Confusão

No problema de classificação estatística, na área do aprendizado de máquina, uma matriz de confusão é uma tabela que permite a visualização do desempenho de um modelo. Cada linha dessa tabela representa as instâncias de uma classe real, enquanto cada coluna ilustra as instâncias de uma classe prevista (Stehman, 1997).

Uma matriz de confusão é composta por quatro valores: verdadeiro positivo, falso negativo, falso positivo e verdadeiro negativo. Essa matriz é muito útil para a avaliação do modelo, uma vez que seus dados descrevem o resultado da classificação de cada registro, além de que é através dela que se torna possível encontrar as demais métricas, como acurácia, precisão, revocação e F1-score.

		PREDITO	
		Classe A	Classe B
VERDADEIRO	Classe A	VP	FN
	Classe B	FP	VN

Figura 6 – Exemplo da matriz de confusão.

Fonte: Adaptado de Hay (1988).

A Figura 6 ilustra um exemplo de uma matriz de confusão. Os quatro valores que

compõem essa tabela são (Ting et al., 2011):

- **Verdadeiro positivo (VP):** indica a quantidade de registros que foram classificados como positivos corretamente, ou seja, a resposta do classificador foi que a entrada era positiva e a entrada realmente era positiva.
- **Verdadeiro negativo (VN):** indica a quantidade de registros que foram classificados como negativos de maneira correta, ou seja, a resposta do classificador foi que a entrada era negativa e a entrada realmente era negativa.
- **Falso positivo (FP):** indica a quantidade de registros que foram classificados como positivos de maneira incorreta, ou seja, a resposta do classificador foi que a entrada era positiva, mas a entrada era negativa.
- **Falso negativo (FN):** indica a quantidade de registros que foram classificados como entradas negativas de maneira incorreta, ou seja, a resposta do classificador foi que a entrada era negativa, mas a entrada era positiva.

2.8.2 Métricas de Qualidade

Com os valores obtidos por meio de uma matriz de confusão, torna-se possível encontrar as demais métricas, como acurácia, precisão, revocação e F1-score.

2.8.2.1 Acurácia

A acurácia é o método mais simples de se calcular, sendo a divisão entre todos os acertos pelo total de exemplos. A Equação 2.9 ilustra como é realizado esse cálculo.

$$\frac{VP + VN}{VP + FP + FN + VN} \quad (2.9)$$

É necessário que o avaliador do modelo tome cuidado ao utilizar essa métrica, uma vez que a quantidade de dados de cada classe deve ser conhecida. Para essa pontuação significar algo, de fato, é necessário que a base de dados que foi utilizada para avaliação tenha uma boa variedade de resultados, não sendo um bom indicador para tarefas que possuam conjunto de dados desbalanceados.

2.8.2.2 Precisão (especificidade)

A precisão, também conhecida como especificidade, indica a relação entre os valores previstos positivos realizados corretamente e todas as previsões positivas, sendo as verdadeiras e as falsas. A Equação 2.10 enfatiza como é realizado esse cálculo.

$$\frac{VP}{VP + FP} \quad (2.10)$$

A métrica precisão fornece a informação sobre falsos positivos, em que se trata de identificar um determinado resultado de maneira precisa. A principal utilização dessa métrica é para modelos onde é preciso minimizar os falsos positivos.

2.8.2.3 Revocação (sensibilidade)

A métrica revocação, também conhecida como sensibilidade, é utilizada para indicar a relação entre as previsões positivas realizadas corretamente e todas as previsões que realmente são positivas (verdadeiro positivo e falso negativo). A Equação 2.11 ilustra como é realizado esse cálculo.

$$\frac{VP}{VP + FN} \quad (2.11)$$

Essa métrica informa sobre os falsos negativos. Ela por si só não é diretamente vinculada a classificar todos os casos corretamente. Ela indica o quanto o modelo está identificando os casos positivos corretamente.

A revocação é bastante útil quando é necessário minimizar os falsos negativos. Ela é especialmente utilizada para casos de diagnósticos, em que pode haver um dano muito maior em não identificar uma doença, do que identificá-la em pacientes saudáveis. Sempre que for preciso minimizar os falsos negativos deve-se buscar uma maior pontuação nessa métrica.

2.8.2.4 F1-score

A métrica F1-score é uma maneira de se visualizar as métricas de precisão e de revocação juntas. O cálculo dessa métrica se dá através da média harmônica, uma vez que os valores calculados são iguais, em que se obtém resultados muito próximos da

média aritmética simples. No entanto, sempre que os valores são diferentes, essa média se aproxima mais dos valores menores. A Equação 2.12 enfatiza como é realizado esse cálculo.

$$\frac{2 \times \textit{precisao} \times \textit{revocacao}}{\textit{precisao} + \textit{revocacao}} \quad (2.12)$$

Em grande parte dos problemas, a F1-score é uma métrica melhor que a acurácia, principalmente em casos que falsos positivos e falsos negativos possuem impactos diferentes para o modelo. Afinal, a F1-score cria um resultado a partir dessas divergências.

2.8.3 Precisão Média e Interseção sobre União

A métrica AP *average precision*, em português, precisão média (por alguns autores associada como *mean Average Precision* (mAP)), é a principal forma de avaliação de métodos de detecção de objetos. Para calculá-la, a precisão média leva em consideração a precisão, revocação e a interseção sobre a união, essa última, em inglês, conhecida como *Intersection over Union* (IoU). A Figura 7 exemplifica como é feito o cálculo da IoU.

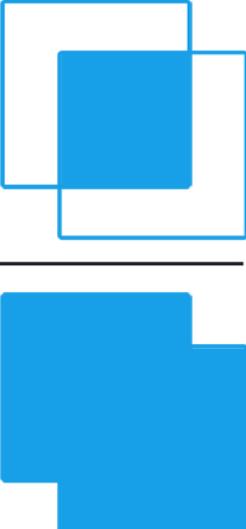
$$\text{IoU} = \frac{\text{Área da interseção}}{\text{Área da união}}$$


Figura 7 – Interseção sobre união.

Fonte: Adaptado de [Rosebrock \(2016\)](#).

A IoU mede a sobreposição entre duas áreas. Essa métrica é utilizada para calcular o quanto a área predita pelo método se sobrepõe a área real, que também pode ser chamada de caixa delimitadora. Em grande parte dos conjuntos de imagens utilizados como *benchmark* para tarefas de detecção de objetos, emprega-se um limite de IoU como

0,5, ao classificar se a previsão realizada é um verdadeiro positivo ou um falso positivo. De forma bem simples, a Figura 8 ilustra três exemplos de caixas delimitadoras reais e previstas.

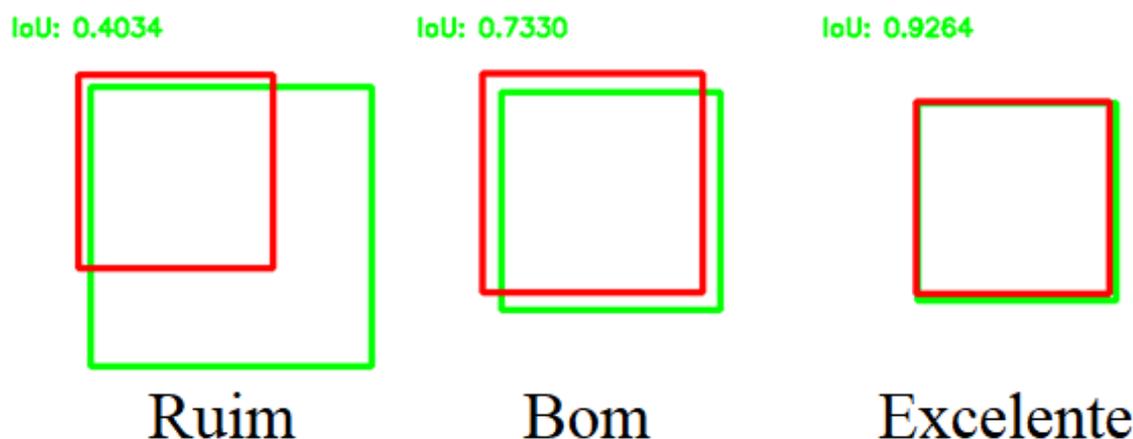


Figura 8 – Exemplos de interseção sobre união.

Fonte: Adaptado de [Rosebrock \(2016\)](#).

Como exemplificado na Figura 8, quanto mais próximo e sobreposto a caixa delimitadora prevista (em vermelho) estiver em relação a caixa delimitadora real (em verde), maior será o valor da IoU. Considerando que a caixa delimitadora prevista esteja completamente fora da área devida, o valor da IoU será 0. Em contrapartida, caso a área prevista esteja sobrepondo exatamente em relação a caixa delimitadora real, o valor da IoU será 1.

2.9 Resumo

Este capítulo descreveu o problema de interesse abordado nessa pesquisa, além de conceituar importantes termos imprescindíveis para a leitura e bom entendimento do trabalho. A utilização do método Kato-Katz na preparação das lâminas para o desenvolvimento do banco de imagens aqui abordado foi de extrema importância, além dos métodos de aprendizado de máquina empregados buscando resolver o problema proposto, em especial as redes neurais convolucionais. Dessa forma, o próximo capítulo tem como objetivo descrever as principais tarefas na área da visão computacional, com foco especial a de detecção de objetos. Além disso, explicar em detalhes alguns dos mais poderosos

métodos de detecção de objetos que podem ser encontrados atualmente na literatura e que são amplamente utilizados tanto na academia quanto na indústria.

Capítulo 3

Detecção de Objetos

Este capítulo tem como objetivo abordar as principais tarefas na área da visão computacional, bem como as principais CNNs empregadas especificamente para resolver cada um dos tipos de tarefas existentes. Além disso, apresentar o funcionamento de maneira detalhada dos principais métodos de detecção de objetos que vem sendo empregados nas aplicações que envolvam essa tarefa.

3.1 Tarefas de Visão Computacional

No campo da visão computacional e do aprendizado profundo existem algumas técnicas que se diferenciam, uma vez que para cada tipo de aplicação é necessário um profundo conhecimento do problema e encontrar a melhor técnica que o resolva. As três mais conhecidas aplicações da visão computacional são: a classificação de imagens, a detecção de objetos e a segmentação de instâncias. A Figura 9 mostra o comparativo entre cada uma delas.

A classificação de imagem é a tarefa de atribuir um rótulo a uma imagem. Geralmente é utilizada quando existe uma única classe que é claramente visível na imagem. A entrada de um modelo classificador é uma imagem com um único objeto, que gera como resultado a definição da classe à qual pertence determinado objeto e o grau de confiança gerado pelo modelo, que representa a probabilidade de o objeto classificado realmente pertencer a aquela classe atribuída.

A localização de objetos é a tarefa de encontrar um determinado objeto na imagem, fazendo com que seja possível encontrar as coordenadas na imagem em que o objeto está

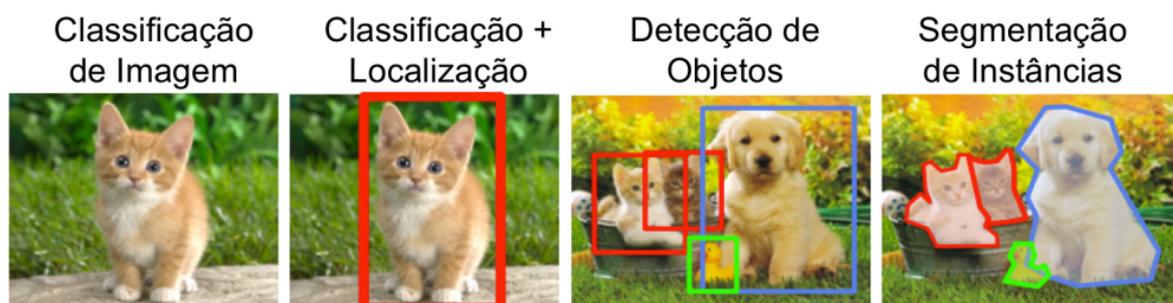


Figura 9 – Comparação entre classificação de imagens, detecção de objetos e segmentação de instâncias.

Fonte: Adaptado de [Li et al. \(2017\)](#).

posicionado e, com isso, desenhar uma caixa delimitadora na imagem. Ao se unir a localização com a classificação, é possível localizar e rotular o objeto, conforme apresentado no segundo exemplo da Figura 9. A limitação que existe na tarefa de localização é que a aplicação exige que se tenha apenas um objeto por imagem, uma vez que possui como resultado as coordenadas de apenas um objeto, em razão de sua saída ser fixa.

A segmentação de instâncias é a tarefa de classificar cada *pixel* que compõe uma imagem. Com isso, é possível agrupar *pixels* e fornecer o contorno exato para cada objeto. Os modelos de segmentação são particularmente úteis em aplicações que necessitam ignorar o plano de fundo em uma imagem, como por exemplo em carros autônomos, em que é fundamental descobrir a localização exata dos demais veículos ao seu redor.

A detecção de objetos é uma técnica de visão computacional que permite identificar e localizar objetos em uma imagem ou vídeo, desenhando caixas delimitadoras em torno dos objetos detectados. Com isso, pode ser utilizada para contar objetos em uma cena, além de determinar e rastrear onde se encontram na imagem, enquanto os classifica com alta precisão. A aplicabilidade dessa técnica pode ser encontrada em diversas áreas, sendo utilizada para detecção facial, de anomalias em imagens médicas, carros autônomos, gestão de obras, entre outras. A Tabela 1 apresenta alguns exemplos de redes neurais convolucionais (CNNs) para visão computacional em suas diferentes tarefas.

Os métodos de detecção de objetos geralmente se enquadram nas abordagens baseadas no aprendizado de máquina ou nas abordagens baseadas no aprendizado profundo. Nas abordagens mais tradicionais utilizando aprendizado de máquina, técnicas de visão computacional são empregadas para examinar os recursos da imagem e, em seguida, utilizar

Tabela 1 – Redes Neurais Convolucionais para Visão Computacional.

CNNs para Visão Computacional						
Classificação de Objetos em Imagens		Detecção de Objetos em Imagens		Segmentação semântica de Objetos em Imagens		Detecção com Segmentação
AlexNet	VGG	R-CNN	SPP-Net	TernausNet	UNet	Mask R-CNN
Google Net	Mobile Net	Fast R-CNN	Yolo			
Dense Net	Squeeze Net	Faster R-CNN	SSD	SegNet	FCN	
ResNet	Inception	RetinaNet	R-FCN			
Determinam se uma imagem contém um determinado objeto		Determinam se a imagem contém um objeto e onde ele se encontra		Determinam se a imagem contém um objeto e quais são os <i>pixels</i> deste objeto		Detectam e segmentam

Fonte: Adaptado de [Wangenheim \(2019\)](#).

um classificador robusto como o SVM para fazer a classificação ([Voulodimos et al., 2018](#); [Chauhan and Singh, 2018](#); [Shinde and Shah, 2018](#); [Chai et al., 2021](#)). Algumas técnicas que empregam essa abordagem mais tradicional são a *Histogram of oriented gradients* (HOG) ([Dalal and Triggs, 2005](#)), *scale-invariant feature transform* (SIFT) ([Lowe, 2004](#)), *Speeded Up Robust Features* (SURF) ([Bay et al., 2006](#)) e a *Viola–Jones object detection framework* baseado em *Haar-like feature* ([Viola and Jones, 2001](#)).

Por outro lado, abordagens baseadas em aprendizado profundo utilizam CNNs para realizar a detecção de objetos sem que seja necessário definir previamente a técnica de extração de recursos, fazendo com que o próprio método durante a etapa de treinamento consiga, automaticamente, definir os valores dos filtros de convolução que melhor se ajustam ao problema. Alguns métodos que empregam a abordagem com CNNs são os que utilizam propostas de região (R-CNN, *Fast R-CNN* e *Faster R-CNN*), *Single Shot MultiBox Detector* (SSD), *You Only Look Once* (YOLO), entre outros.

Um modelo de detecção de objetos é treinado para detectar a presença e a posição na imagem de diferentes classes de objetos. Por exemplo, no treinamento de imagens que contém diferentes frutas, o modelo é capaz de aprender a identificar maçãs, bananas, morangos e a respectiva posição de cada uma dessas frutas na imagem. Quando é fornecida uma nova imagem em um modelo treinado, ele exibe uma lista dos objetos que detectou,

o local da caixa delimitadora que contém o objeto e uma pontuação, também chamada de escore de confiança, que indica a confiança da detecção estar correta. Em um exemplo hipotético de uma imagem que contém algumas frutas, a saída do modelo pode ser exemplificada na Tabela 2.

Tabela 2 – Exemplo hipotético da saída gerada por um detector de objetos.

Classe	Pontuação	Localização
maçã	0.92	[18, 21, 57, 63]
banana	0.88	[100, 30, 180, 150]
morango	0.87	[7, 82, 89, 163]
banana	0.23	[42, 66, 57, 83]
maça	0.11	[6, 42, 31, 58]

Para a interpretação dos resultados de um detector de objetos é necessário que se avalie a pontuação e a localização de cada objeto detectado. A pontuação é um número que varia entre zero a um, indicando a confiança que o modelo tem em dizer que realmente a detecção está correta. Quanto mais próximo de um, mais confiante está o modelo. Dependendo da aplicação é necessário que se defina um limiar de corte na pontuação para que o modelo ignore alguns resultados. Para o exemplo hipotético, caso o limiar de corte do escore de confiança fosse 0.5 (50%), o modelo descartaria os dois últimos resultados da Tabela 2.

O limiar adotado para o escore de confiança é específico para cada aplicação e definido através da realização de uma bateria de testes. É necessário que se leve em consideração os falsos positivos (objetos ou áreas da imagem que são identificados incorretamente, uma vez que não estão presentes na imagem) e os falsos negativos (objetos ou áreas que deixam de ser identificados pelo fato de que o limiar do escore de confiança esteja baixo).

Além da verificação do valor encontrado para o escore de confiança, outro resultado gerado pela saída de um detector de objetos é uma matriz que contém n (quantidade de objetos detectados) linhas e quatro colunas, que representa um retângulo delimitador que envolve a posição de cada objeto. Existem dois formatos possíveis para a configuração da caixa delimitadora, o COCO (x-esquerda superior, y-esquerda superior, largura, altura) e o Pascal VOC (x-esquerda superior, y-esquerda superior, x-direita, y-direita). Os valores encontrados são representados sendo a distância medida em *pixels* para cada borda do retângulo, em relação ao canto da imagem, para cada eixo.

Cada método de detecção de objetos, seja utilizando a abordagem de aprendizado

de máquina ou a abordagem empregando o aprendizado profundo, tem suas especificidades com vantagens e desvantagens, sendo que seus resultados e suas aplicações variam de acordo com o problema. Atualmente, de uma forma geral, três deles vem se destacando com uma maior acurácia e um menor tempo de detecção, que são a evolução dos métodos baseados em região, que hodiernamente se concentra no *Faster R-CNN*, e os métodos de uma etapa, YOLO e SSD.

3.2 R-CNN

Proposto por [Girshick et al. \(2014\)](#), o R-CNN revolucionou o campo da visão computacional no que tange a detecção de objetos, que até o seu desenvolvimento em 2014, consistiam-se na abordagem de força bruta, varrendo toda a imagem em janelas da esquerda para a direita e de cima para baixo. Com as janelas selecionadas, as partes menores da imagem eram mostradas para um classificador com o objetivo de identificar os objetos ali presentes.

O principal problema das abordagens desenvolvidas até o surgimento do R-CNN era o grande número de imagens que eram submetidas ao classificador, uma vez que o objeto poderia se encontrar em qualquer ponto da imagem e por isso era selecionado o número máximo de janelas possíveis, sendo limitadas pelo poder computacional disponível no momento.

Buscando resolver esse problema, [Girshick et al. \(2014\)](#) desenvolveram o R-CNN, que se trata de uma rede neural convolucional baseada em regiões. A ideia principal do método pode ser resumida em três módulos: primeiro, são geradas regiões independentes de categorias usando alguma heurística previamente definida, em seguida, usa-se uma CNN que extrai as características das regiões e, por último, é utilizado um classificador a partir das características extraídas. A Figura 10 retirada do artigo original de [Girshick et al. \(2014\)](#) ilustra a arquitetura do R-CNN.

Conforme pode ser observado na Figura 10, a seleção das regiões propostas é realizada após a entrada da imagem no método. Essa seleção não é feita de maneira aleatória ou por força bruta conforme feita pelos métodos desenvolvidos anteriormente. O R-CNN utiliza a pesquisa seletiva para contornar o problema da seleção de muitas regiões candidatas, dessa vez, gerando aproximadamente duas mil propostas.

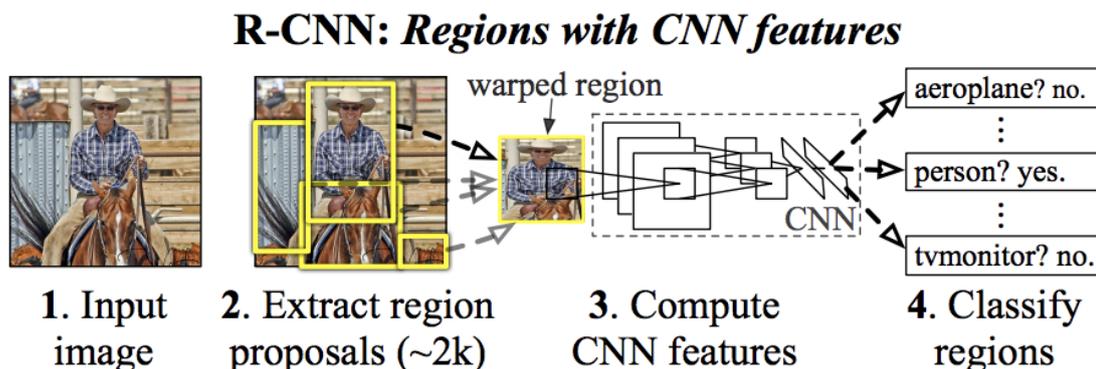


Figura 10 – Arquitetura do R-CNN.

Fonte: Girshick et al. (2014).

A pesquisa seletiva para o reconhecimento de imagens foi introduzida no trabalho de Uijlings et al. (2013), que combina heurísticas de força bruta com segmentação exaustiva. No exemplo da Figura 11 é possível definir algumas premissas da aplicação da pesquisa seletiva para a segmentação de imagens perceptíveis ao olho humano que podem ser adotadas para a separação de determinados objetos.

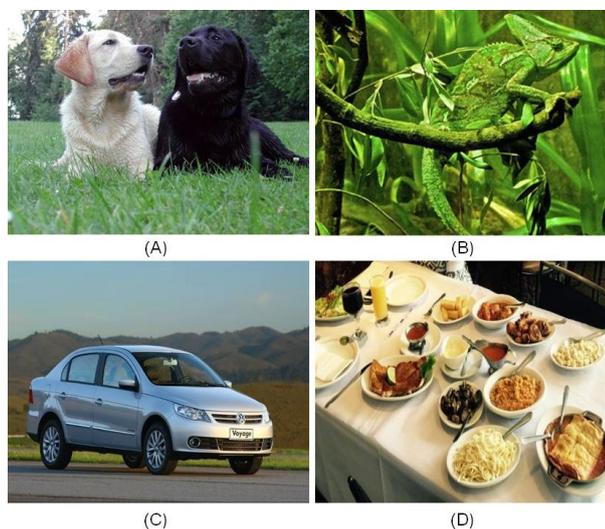


Figura 11 – Exemplos de aplicação da pesquisa seletiva.

A Figura 11 (A) mostra dois cachorros que podem ser facilmente distinguidos por causa da coloração de cada um. Na Figura 11 (B) é possível separar o camaleão dos demais objetos da imagem devido sua textura. A Figura 11 (C) ilustra um automóvel que pode ter suas rodas distinguidas do restante da imagem por ser um objeto fechado. Já para a Figura 11 (D) seria necessário a combinação dessas técnicas para que se consiga

separar cada objeto do exemplo.

O algoritmo de pesquisa seletiva funciona da seguinte forma: primeiro é gerado uma segmentação parcial da imagem de entrada utilizando o método descrito por [Felzenszwalb and Huttenlocher \(2004\)](#). Em seguida, é feita recursivamente uma combinação das regiões semelhantes, fazendo com que regiões menores se agrupem de modo que se tornem regiões maiores. Para o agrupamento dessas regiões é utilizado o algoritmo de *Greedy* ([Edmonds, 1971](#)) que, em resumo, compara iterativamente a semelhança entre cada região e as combina. A pesquisa seletiva considera quatro tipos de similaridades ao realizar as combinações de regiões menores em regiões maiores, são elas ([Uijlings et al., 2013](#)):

- **Cor:** Para calcular a similaridade de cor é gerado o histograma para cada região de cada canal de cor presente na imagem.
- **Textura:** A similaridade de textura é calculada utilizando oito derivadas gaussianas da imagem, extraindo o histograma para cada canal de cor.
- **Tamanho:** A ideia básica da similaridade de tamanho é fazer com que regiões menores se agrupem facilmente, uma vez que caso apenas regiões maiores sejam combinadas, as propostas de regiões serão geradas apenas em determinados locais.
- **Preenchimento:** Essa similaridade mede quão bem duas regiões se encaixam, levando em consideração a distância entre regiões.

Por último, são utilizadas as regiões segmentadas propostas para gerar os locais dos objetos candidatos. A Figura 12 ilustra esse processo exemplificando o funcionamento para cada iteração que ocorre no algoritmo de pesquisa seletiva.

Tendo sido realizada a pesquisa seletiva e de posse de aproximadamente duas mil regiões candidatas, a imagem é cortada conforme as regiões candidatas e posteriormente redimensionadas para um tamanho fixo de modo que seja compatível com a arquitetura da CNN utilizada, que para o trabalho de [Girshick et al. \(2014\)](#), foi a AlexNet ([Krizhevsky et al., 2012](#)). A arquitetura da AlexNet consiste em oito camadas: cinco camadas convolucionais e três camadas totalmente conectadas, conforme é ilustrado na Figura 13 retirada do artigo original desenvolvido por [Krizhevsky et al. \(2012\)](#).

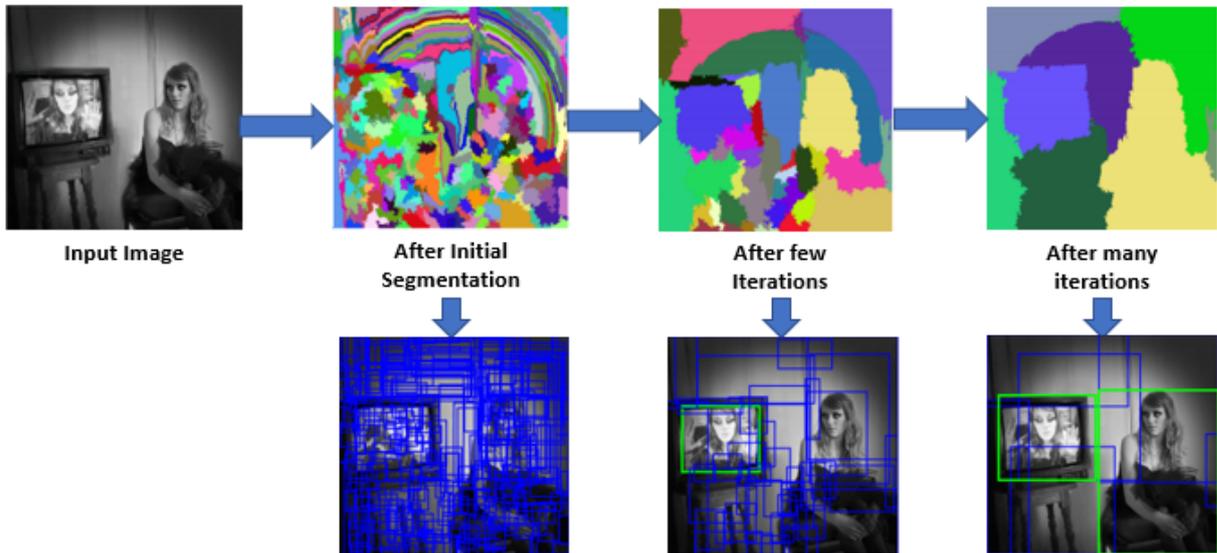


Figura 12 – Exemplo de funcionamento do algoritmo de pesquisa seletiva.

Fonte: Saxena (2021).

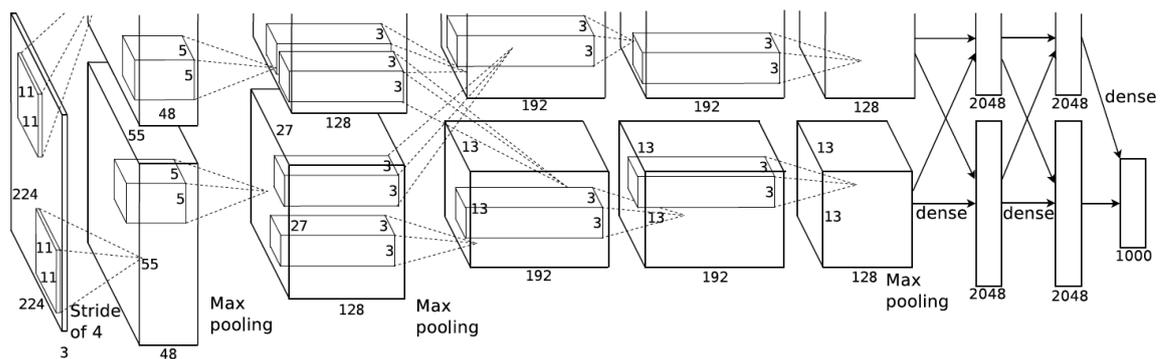


Figura 13 – Arquitetura da AlexNet.

Fonte: Krizhevsky et al. (2012).

Para o método R-CNN proposto por Girshick et al. (2014), não foi utilizada a arquitetura completa da AlexNet. O classificador que é composto pelas três camadas totalmente conectadas foi substituído pelo classificador SVM (*support vector machine*), que encontra um hiperplano em um espaço N-dimensional com o objetivo de classificar diferentes classes representadas por pontos distintos (Chang and Lin, 2011). Sendo assim, a parte utilizada da arquitetura da AlexNet foram as camadas de convolução, as quais são responsáveis por extrair as características mais pertinentes de cada uma das duas mil imagens (regiões) candidatas.

Com as duas mil imagens candidatas sendo submetidas, uma a uma, junto às

camadas convolucionais da arquitetura pré-treinada da AlexNet, é possível que cada uma dessas regiões seja classificadas pela SVM, obtendo-se como resultado a presença ou não de determinado objeto naquela região proposta. A partir daí, tem-se as partes da imagem que possivelmente contém algum objeto, dado um limiar de confiança previamente especificado.

Além de prever a presença ou não de um objeto em determinada região candidata, é utilizado um regressor linear (usando camadas totalmente conectadas) para prever os valores de deslocamento e ajuste da posição da caixa delimitadora para cada objeto, com o objetivo de aumentar a precisão da localização na detecção. Por fim, é empregada a técnica de supressão não máxima tentando evitar a detecção repetida de uma mesma instância, que ocorre na situação em que o modelo encontra várias caixas delimitadores para o mesmo objeto.

O método R-CNN revolucionou o campo da visão computacional superando os algoritmos clássicos de detecção de objetos até ali desenvolvidos. Apesar disso, apresenta algumas desvantagens:

- É necessário muito tempo para se treinar a rede, uma vez que o método propõe duas mil regiões candidatas por imagem.
- Não pode ser implementado em aplicações que se exige processamento em tempo real, pois leva cerca de 49 segundos para cada imagem de teste, considerando os testes detalhados no trabalho de [Girshick et al. \(2014\)](#).
- O algoritmo de pesquisa seletiva é fixo e não treinado, ou seja, pode ocorrer a situação em que haja várias propostas de regiões candidatas ruins.
- É necessário que se execute a extração de características para cada uma das duas mil regiões propostas por imagem.
- A extração de recursos, o SVM e o regressor linear são executados em momentos diferentes, ou seja, envolve três fluxos independentes, sem nenhuma computação compartilhada.

3.2.1 Fast R-CNN

Apesar de cumprir muito bem o que se propõe a fazer, o R-CNN apresenta algumas desvantagens que são um gargalo no campo da visão computacional para a detecção de

objetos. Com isso, o mesmo autor desenvolveu um novo método que apresenta algumas vantagens em relação ao anterior, como o aumento da velocidade na etapa de treinamento e teste, além da precisão na detecção. Girshick (2015) o denominou como *Fast R-CNN*.

A abordagem do método *Fast R-CNN* é semelhante a utilizada no R-CNN, mas, dessa vez, as três partes que compunham o sistema do R-CNN (CNN, SVM e o regressor da caixa delimitadora) foram combinadas em apenas uma arquitetura, fazendo com que fosse possível utilizar os recursos da computação compartilhada. Ao invés de alimentar a CNN com as duas mil propostas de regiões geradas pelo algoritmo de pesquisa seletiva, essa é submetida apenas a imagem original para o extrator de recursos, de modo que otimize bastante o tempo de processamento nessa operação.

A Figura 14 retirada do artigo original de Girshick (2015) ilustra a arquitetura proposta para o *Fast R-CNN*. Uma imagem de entrada e aproximadamente duas mil regiões de interesse (RoIs) são submetidas a uma CNN. Cada RoI é agrupada em um mapa de recursos de tamanho fixo, para que em seguida seja transformado em um vetor de recursos que será inserido em camadas totalmente conectadas. A arquitetura da rede possui dois vetores de saída para cada RoI: as probabilidades geradas por meio da função de ativação *softmax* (antes feita pelo SVM no R-CNN) e os valores de ajuste de precisão da caixa delimitadora para cada classe.

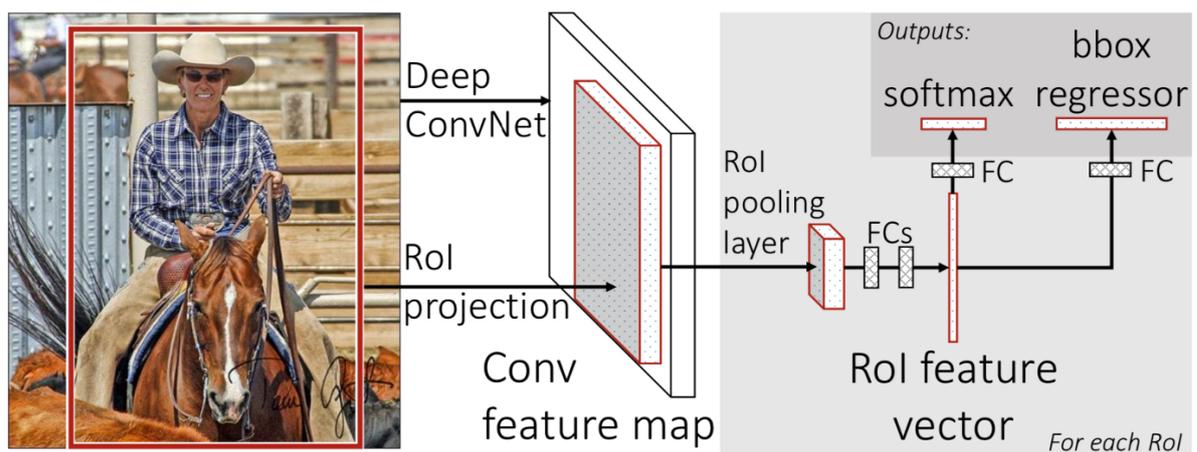


Figura 14 – Arquitetura do *Fast R-CNN*.

Fonte: Girshick (2015).

A principal diferença entre a arquitetura do R-CNN e do *Fast R-CNN* está na inserção de uma camada de *pool* de RoI na última camada da CNN. Seu principal objetivo é,

para todas as regiões candidatas encontradas pela pesquisa seletiva, descobrir suas entradas correspondentes do mapa de recursos e as dimensioná-las para um tamanho predefinido, de modo que seja compatível com as posteriores camadas totalmente conectadas.

Com a camada de *pool* de RoI é possível se obter rapidamente um vetor dos mapas de recursos correspondentes com um tamanho fixo, mesmo que se tenha milhares de regiões candidatas com tamanhos diferentes. Com isso, aumenta-se a velocidade do processamento da operação, uma vez que computar as convoluções nos estágios iniciais do processamento é considerado caro, fazendo-se com que a empregabilidade de tal abordagem economize tempo. A Figura 15 apresenta um exemplo de aplicação do *pool* de RoI em um mapa de recursos.

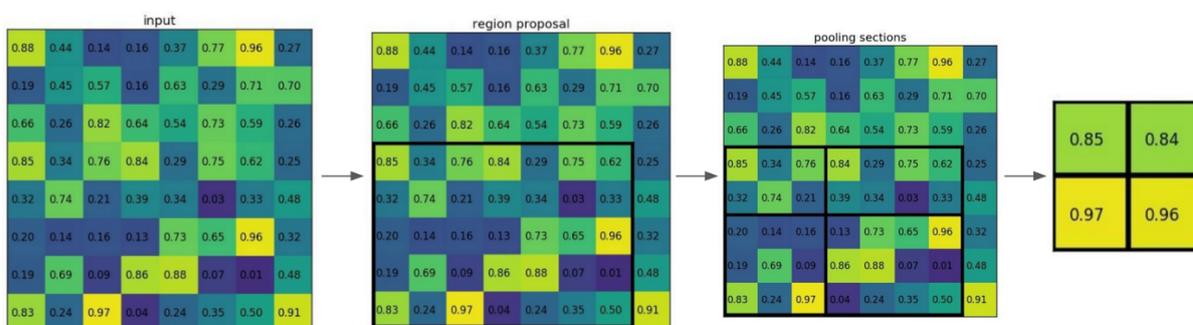


Figura 15 – Exemplo da aplicação do *pool* de RoI.

Fonte: Girshick (2015).

Para cada proposta de região, é extraída a parte correspondente do mapa de recursos. Em seguida, é encontrada a localização que coincide entre a proposta de região e o mapa de recursos, para que essa correspondência seja redimensionada para uma estrutura de tamanho fixo. Com isso, é possível se obter um vetor de recursos que sempre tem o mesmo tamanho, permitindo que se reutilize o mapa de recursos obtidos pela CNN.

Finalmente, tendo obtido o vetor de recursos, é possível alimentar as camadas totalmente conectadas. A primeira camada de classificação possui uma função de ativação *softmax* que tem como objetivo descobrir a probabilidade para qual classe pertence o objeto encontrado. A segunda camada, que é o regressor da caixa delimitadora, é responsável por ajustar e prever o deslocamento das coordenadas da caixa delimitadora para cada classe de objeto.

A principal razão que faz o *Fast* R-CNN ser mais rápido do que o R-CNN é o fato de que no *Fast* R-CNN não é necessário alimentar a CNN com duas mil propostas de

regiões candidatas para cada imagem. Ao invés disso, a operação de convolução é realizada somente uma vez para cada imagem. Além disso, no *Fast R-CNN* é utilizado o conceito de computação compartilhada, uma vez que as três partes que compõem o sistema do R-CNN de forma independente (CNN, SVM e o regressor da caixa delimitadora), agora estão em apenas uma única arquitetura.

3.2.2 Faster R-CNN

Os métodos de detecção de objetos R-CNN e *Fast R-CNN* utilizam a pesquisa seletiva para descobrir as propostas de região. A pesquisa seletiva demanda um alto poder computacional e é extremamente lenta, afetando o desempenho dos métodos que a empregam. Por isso, [Ren et al. \(2015\)](#) propôs um novo método de detecção de objetos que elimina a pesquisa seletiva e adiciona uma rede de proposta de região (RPN, do inglês *Region Proposal Network*) para executar a tarefa de descobrir as regiões candidatas. O novo método foi chamado pelos seus criadores de *Faster R-CNN* ([Ren et al., 2015](#)).

A Figura 16 retirada do trabalho de [Ren et al. \(2015\)](#) apresenta a arquitetura do método *Faster R-CNN*. É possível perceber a semelhança com o método *Fast R-CNN*, exceto na etapa de proposta de região, em que a pesquisa seletiva foi substituída por uma RPN.

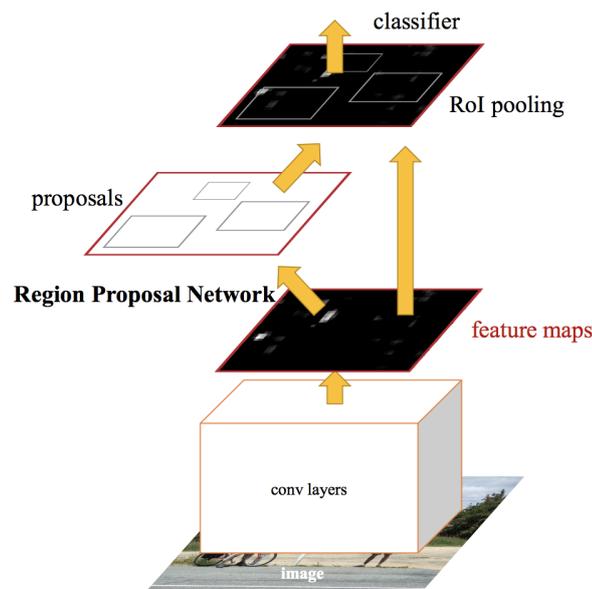


Figura 16 – Arquitetura do *Faster R-CNN*.

Fonte: [Ren et al. \(2015\)](#).

Em resumo, o método R-CNN descobre as propostas de região no nível de *pixel* da imagem, enquanto o *Fast R-CNN* no nível do mapa de recursos obtidos pela CNN, ambos utilizando a pesquisa seletiva. Assim, a proposta de descobrir as regiões candidatas são dissociadas com o restante do sistema, para que os erros obtidos nessa etapa não possam ser retropropagados. Ao se empregar uma RPN, conforme adotado pelo *Faster R-CNN*, todos os recursos são compartilhados pelo sistema de detecção, aumentando a velocidade e melhorando a precisão do método.

A primeira etapa realizada pelo método *Faster R-CNN* é a extração de características de cada imagem a partir das camadas de convolução de uma CNN pré-treinada. Em seguida, uma janela deslizante é utilizada pela RPN em cada local do mapa de recursos obtido. A Figura 17 retirada do trabalho de Ren et al. (2015) demonstra o funcionamento da RPN.

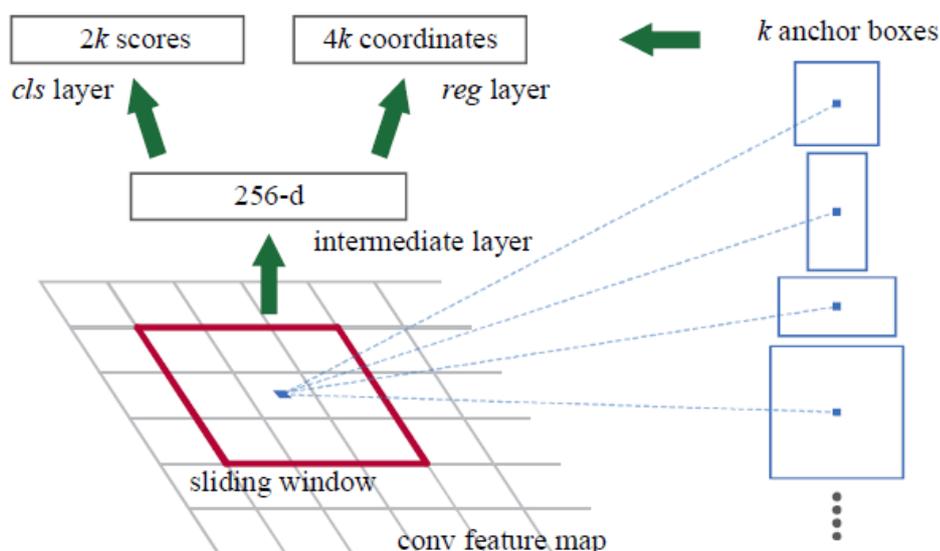


Figura 17 – Funcionamento da RPN.

Fonte: Ren et al. (2015).

Para cada local do mapa de características são utilizadas caixas de ancoragem para gerar as propostas de região. Uma âncora é o ponto central da janela deslizante em que, para cada local, são produzidas nove propostas de regiões baseadas em escalas e proporções, conforme exemplificado na parte direita da Figura 17. Com isso, para cada região candidata, tem-se um valor de probabilidade que representa a chance de existência de um determinado objeto e suas respectivas coordenadas, para que se decida o descarte ou não daquela proposta de região.

Assim, a RPN realiza a verificação prévia sobre qual local contém um objeto para que, posteriormente, as outras partes restantes do método de detecção de objetos *Faster R-CNN* fiquem responsáveis por definir a qual classe o suposto objeto pertence e encontrar os valores de ajuste a partir do regressor, com o objetivo de aumentar a precisão da caixa delimitadora, assim como é feito no *Fast R-CNN*.

Os métodos de detecção de objetos R-CNN e *Fast R-CNN* são considerados lentos e não podem ser utilizados em problemas que exijam a detecção de objetos em tempo real. Um dos motivos que causa esse processamento demorado é que a proposta de regiões candidatas para esses métodos é realizada pelo algoritmo de pesquisa seletiva. A substituição desse algoritmo para um mais sofisticado foi primordial para o aumento na velocidade de detecção do método *Faster R-CNN*, permitindo que o mesmo possa ser utilizado em aplicações que exijam o processamento em tempo real. O gráfico ilustrado na Figura 18 apresenta a comparação de velocidade do tempo de teste para cada imagem que é alimentada em cada um dos métodos de detecção de objetos baseados em região. Os detalhes dos experimentos podem ser encontrados no trabalho de [Girshick et al. \(2014\)](#).

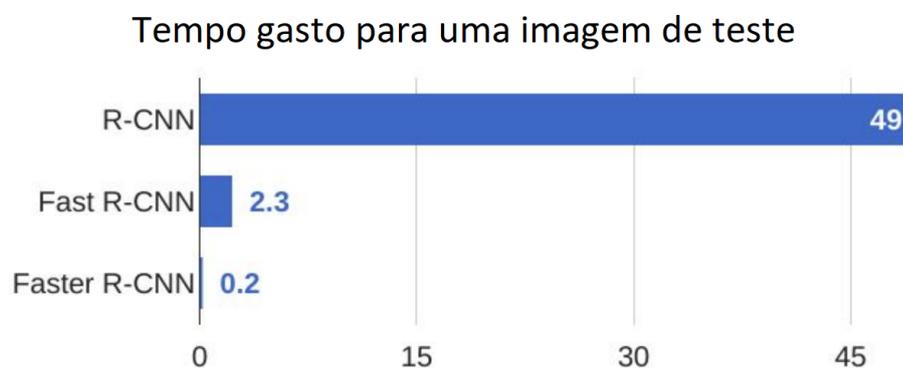


Figura 18 – Comparativo entre a velocidade do tempo de inferência para cada um dos métodos de detecção de objetos baseados em regiões.

Fonte: Adaptado de [Gandhi \(2018\)](#).

3.3 YOLO

Até o desenvolvimento do método de detecção de objetos *You Only Look Once* (YOLO) ([Redmon et al., 2016](#)), em português “você só olha uma vez”, o que se tinha eram métodos baseados no aprendizado de máquina e métodos baseados em propostas de regiões utilizando a abordagem de aprendizado profundo, tal como os métodos da família

R-CNN, que necessitam executar duas etapas para processar uma imagem, impactando diretamente em sua velocidade, tornando o algoritmo lento em aplicações que demandam detecções em tempo real.

O método desenvolvido no trabalho de [Redmon et al. \(2016\)](#) sugere uma rede totalmente convolucional e unificada de modo que possa executar todas as etapas uma única vez. Além disso, tem como característica ser uma rede de treinamento de ponta a ponta, ou seja, o erro obtido na camada de saída é retropropagado por toda a rede. A Figura 19 retirada do artigo original de [Redmon et al. \(2016\)](#) ilustra o modelo YOLO, que enxerga a detecção como um problema de regressão.

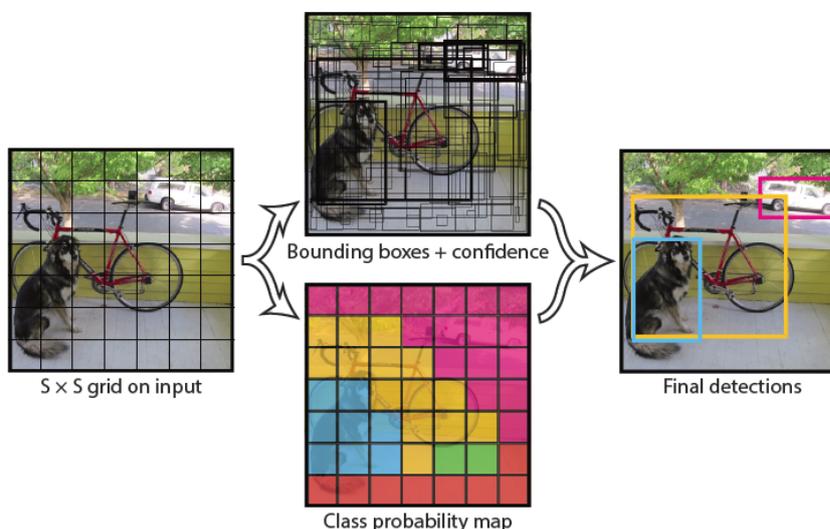


Figura 19 – Modelo do método YOLO.

Fonte: [Redmon et al. \(2016\)](#).

A imagem de entrada é dividida em uma grade $S \times S$ ($S=7$), conforme apresentada no canto esquerdo da Figura 19. Caso o centro de um objeto esteja posicionado em alguma célula da grade, essa mesma célula se torna responsável por detectar esse objeto. Cada uma das 49 células da grade prevê B caixas delimitadoras ($B=2$) e pontuações de confiança que representam a chance de algum objeto estar presente na caixa gerada. Cada caixa delimitadora é composta por cinco valores, que são as coordenadas x e y que representam o centro da caixa em relação aos limites da célula da grade, a altura e largura em relação à imagem inteira e o escore de confiança que representa a intersecção sobre a união (IOU) entre a caixa prevista e a caixa gerada aleatoriamente no início do treinamento.

Além da imagem de entrada que é dividida em uma grade 7×7 e a geração de

duas caixas delimitadoras e suas cinco previsões, cada célula da grade também prevê probabilidades condicionais para cada classe do problema. No desenvolvimento do método YOLO a proposta inicial continha 20 classes de diferentes objetos (Redmon et al., 2016), resultando em um tamanho de saída da rede em $7 \times 7 \times (2 \times 5 + 20) = 1470$. A Figura 20 ilustra a saída gerada pelo método YOLO.

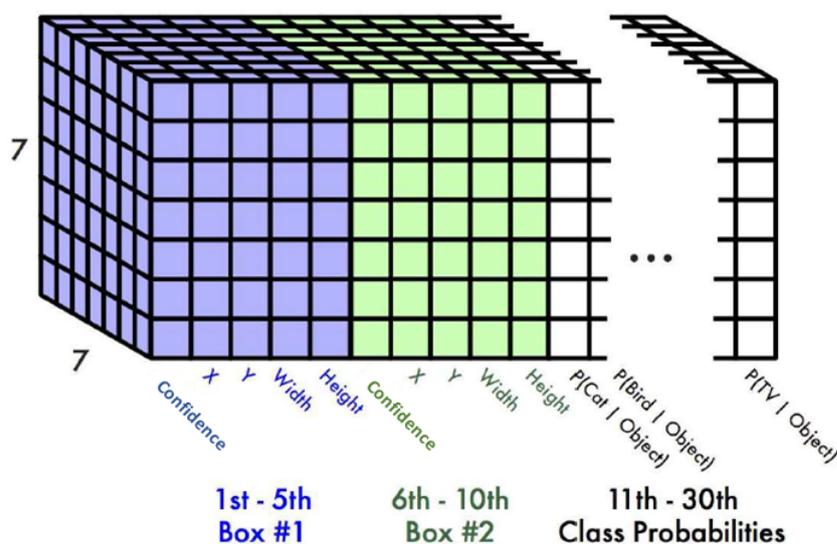


Figura 20 – Saída do método YOLO.

Fonte: Redmon et al. (2016).

A arquitetura do método YOLO foi baseada na arquitetura do classificador GoogleNet (Szegedy et al., 2015), composto por 24 camadas convolucionais e duas camadas totalmente conectadas. As camadas de convolução utilizam extratores de características de tamanho 1×1 para reduzir o número de parâmetros. A Figura 21 retirada do artigo original de Redmon et al. (2016) apresenta a arquitetura do detector de objetos proposto em sua primeira versão.

Portanto, é possível perceber na Figura 22 que ilustra o pipeline do método YOLO que a imagem de entrada passa pela rede uma única vez e, em seguida, já é possível detectar os objetos, fazendo com que se tenha um aprendizado de ponta a ponta. Isso torna o método extremamente rápido, podendo ser aplicado em problemas que exigem processamento em tempo real, superando em velocidade os detectores de objetos baseados em regiões.

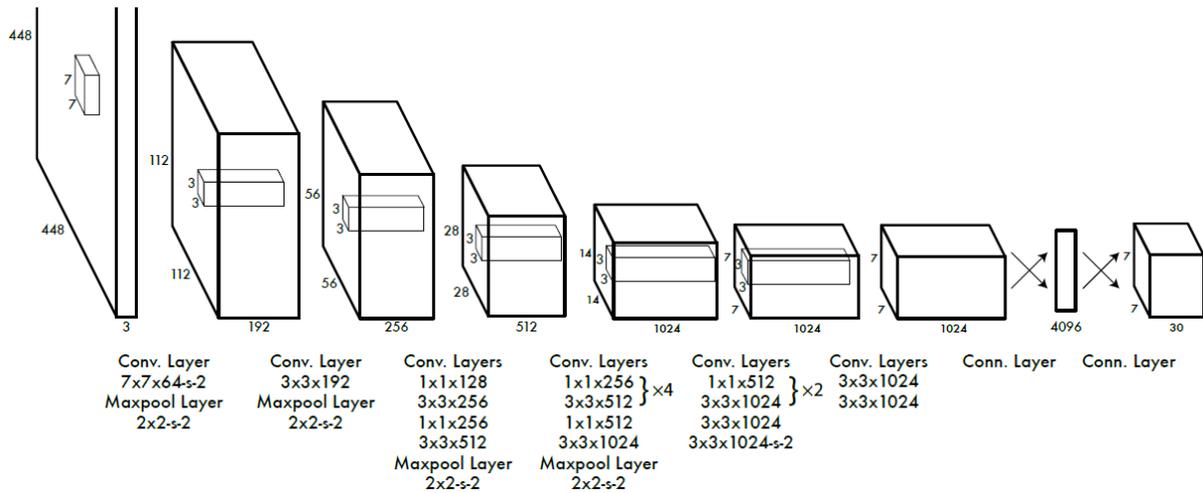


Figura 21 – Arquitetura do método YOLO.

Fonte: Redmon et al. (2016).

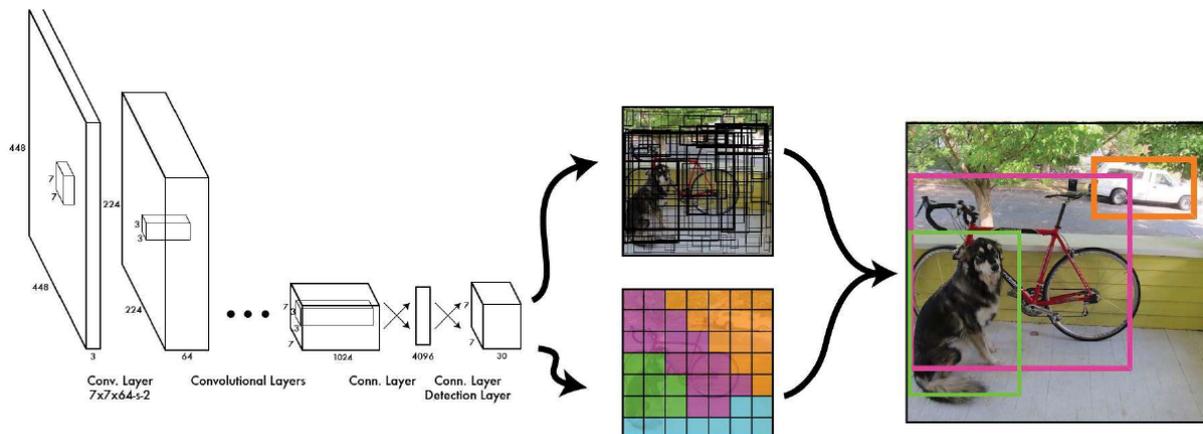


Figura 22 – Pipeline do método YOLO.

Fonte: Redmon et al. (2016).

3.3.1 YOLOv2

Apesar do sucesso do método YOLO desenvolvido no trabalho de Redmon et al. (2016), a proposta ainda tinha alguns problemas, deixando margem para melhorias. Até o momento, o valor de mAP estava bem abaixo em relação aos detectores de objetos baseados em regiões. Assim sendo, um ano depois foi desenvolvido pelos mesmos autores o YOLOv2 (Redmon and Farhadi, 2017), que assim como definido por eles no próprio título do trabalho, é um método “melhor, mais rápido e mais forte”.

Foi adicionada em todas as camadas convolucionais a normalização em lote, que consiste em normalizar os dados fornecidos a cada camada oculta, produzindo um efeito regularizador no treinamento e consequentemente diminuindo o tempo do treino. Além

disso, o YOLOv2 não possui as camadas totalmente conectadas, as quais foram substituídas por caixas de ancoragem, de modo a prever as caixas delimitadoras. Outra melhoria em relação a primeira versão do YOLO foi a adição do clusterizador K-means no processo, que agora é responsável por encontrar as melhores caixas de ancoragem baseadas nas anotações das imagens de treinamento, de forma que essas propostas de âncoras não sejam mais aleatórias, conforme feitas no método anterior. Todas essas melhorias foram essenciais para que o método conseguisse alcançar um considerável valor de mAP, assim como seus concorrentes.

Além de todas as melhorias realizadas que contribuíram significativamente para um melhor valor de mAP, outro fator que favoreceu tal melhora foi a modificação da arquitetura da rede. O detector de objetos YOLOv2 utiliza a estrutura Darknet-19 (Redmon, 2016) para a extração de recursos, o qual possui mais convoluções 1x1, reduzindo ainda mais o número de parâmetros e tornando o método ainda mais rápido que o anterior. A Figura 23 apresenta a arquitetura da Darknet-19.

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

Figura 23 – Arquitetura da estrutura Darknet-19.

Fonte: Redmon (2016).

Além de “melhor” e “mais rápido”, Redmon and Farhadi (2017) conseguiram em seu trabalho um método “mais forte”, que foi embasado na combinação entre conjuntos

de dados de classificação e detecção. O conjunto de imagens Microsoft COCO (Lin et al., 2014) e ImageNet (Deng et al., 2009) foi relacionado através de uma árvore hierárquica, permitindo assim, que o YOLOv2 agora possa ser capaz de detectar 9418 diferentes classes de objetos, justificando o título do artigo “YOLO9000: Better, Faster, Stronger”.

3.3.2 YOLOv3

Apesar de todas as melhorias propostas no YOLOv2, em que se alcançou resultados excelentes, os autores continuaram a trabalhar no método e conseguiram obter resultados ainda mais satisfatórios, os quais foram publicados por Redmon and Farhadi (2018). Foram feitas algumas melhorias incrementais nessa terceira versão, sendo que a principal delas foi a substituição da Darknet-19 pela Darknet-53, implicando em uma arquitetura de rede um pouco maior. Além disso, não é mais utilizada a função de ativação *softmax* na saída, tendo sido substituída pela função de ativação logística. Foi empregado um extrator de recursos mais sólido e melhor, adotando técnicas de *upsampling* e concatenação (Shan et al., 2008).

Para a previsão da caixa delimitadora, a técnica utilizada é a mesma que na versão anterior, exceto pelo fato de que foi adicionado ao YOLOv3 uma pontuação de objetividade para a previsão das caixas, com o objetivo de melhorar a escolha da caixa utilizada nos casos em que objetos se sobrepõem uns aos outros. Essa pontuação de objetividade é prevista por meio da regressão logística, levando em consideração a compensação dos centroides encontrados pelo algoritmo K-means. A Figura 24 retirada do trabalho original de Redmon and Farhadi (2018) demonstra como é feito o processo da previsão das caixas delimitadoras no YOLOv3.

Para a classificação dos objetos, o YOLOv3 deixou de utilizar a função de ativação *softmax* porque os autores concluíram com os testes realizados que a função era desnecessária. Após a remoção, conseguiu-se melhorar o desempenho do processo. Como alternativa, foram empregados vários classificadores logísticos independentes. Com isso, mesmo que o YOLOv3 fosse movido para domínios mais complexos, como por exemplo para o caso de conjuntos de imagens como o *Open Images Dataset*, o método de detecção conseguiria obter um desempenho satisfatório, justamente por causa dessa mudança efetuada.

Ao contrário do YOLOv2, o detector de objetos YOLOv3 extrai os recursos de

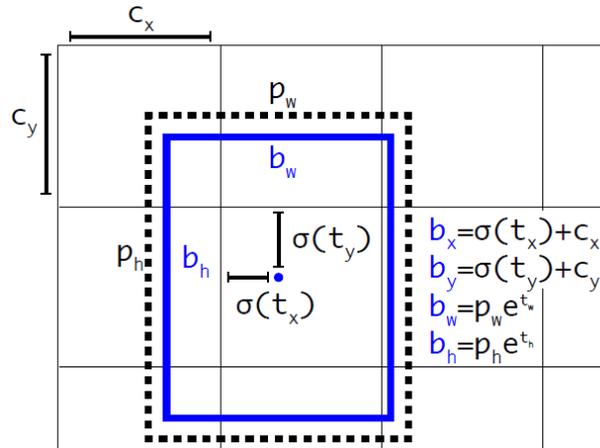


Figura 24 – Previsão da caixa delimitadora.

Fonte: Redmon and Farhadi (2018).

três escalas diferentes, concatenando-os no final. Essa configuração permite que o método obtenha informações semânticas mais significativas e refinadas do mapa de recursos. Além disso, foram adicionadas algumas camadas convolucionais para que se processe o mapa de recursos combinado e, eventualmente, fazer a previsão da saída, a qual é duas vezes maior do que a utilizada no método anterior. Importante ressaltar que o algoritmo K-means também é utilizado com o objetivo de encontrar as coordenadas da caixa delimitadora.

Por fim, o YOLOv3 empregou uma nova rede para realizar a extração de recursos. A estrutura Darknet-19 utilizada na versão anterior, agora foi substituída pela Darknet-53, que é mais profunda e é composta por 53 camadas convolucionais. Assim como no YOLOv2, a versão aprimorada também continuou usando a normalização em lote em suas camadas de convolução. A Figura 25 retirada do artigo original de Redmon and Farhadi (2018) ilustra a arquitetura da Darknet-53.

3.4 SSD

No trabalho publicado por Liu et al. (2016) foi apresentado o método *SSD: Single Shot MultiBox Detector*, em português, detector *MultiBox* de disparo único. No SSD, é necessária apenas uma passagem pela rede para que o método consiga detectar os objetos na imagem, assim como acontece no YOLO. O novo detector proposto tem seu funcionamento diferente de como acontece nas abordagens baseadas na rede de proposta de regiões, que necessitam de duas imagens para realizar as detecções, uma para gerar as

	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	$3 \times 3 / 2$	128×128
1x	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			128×128
	Convolutional	128	$3 \times 3 / 2$	64×64
2x	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Residual			64×64
	Convolutional	256	$3 \times 3 / 2$	32×32
8x	Convolutional	128	1×1	
	Convolutional	256	3×3	
	Residual			32×32
	Convolutional	512	$3 \times 3 / 2$	16×16
8x	Convolutional	256	1×1	
	Convolutional	512	3×3	
	Residual			16×16
	Convolutional	1024	$3 \times 3 / 2$	8×8
4x	Convolutional	512	1×1	
	Convolutional	1024	3×3	
	Residual			8×8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figura 25 – Arquitetura da estrutura Darknet-53.

Fonte: [Redmon and Farhadi \(2018\)](#).

propostas candidatas e outra para detectar o objeto em cada região.

O título do artigo *SSD: Single Shot MultiBox Detector* de [Liu et al. \(2016\)](#) foi escolhido propositalmente para ressaltar as vantagens que o método oferece. *Single Shot* significa que tanto as operações de localização quanto classificação de objetos são realizadas em uma única passagem seguindo pela rede. *MultiBox* é o nome da técnica desenvolvida no trabalho de [Szegedy et al. \(2014\)](#) que é utilizada para regressão da caixa delimitadora. *Detector* caracteriza o fato de que a arquitetura proposta, além de classificar os objetos, também os detecta.

A arquitetura do método SSD pode ser dividida em três componentes principais: uma rede base, camadas extras para extração de recursos e camadas de previsão. A rede base é composta por camadas de convolução utilizadas em qualquer rede padrão de classificação de imagens, pré-treinada em algum conjunto de dados. [Liu et al. \(2016\)](#) baseou-se na rede VGG-16, substituindo as camadas totalmente conectadas por camadas convolucionais. A Figura 26 retirada do artigo original de [Liu et al. \(2016\)](#) ilustra a arquitetura do método

SSD.

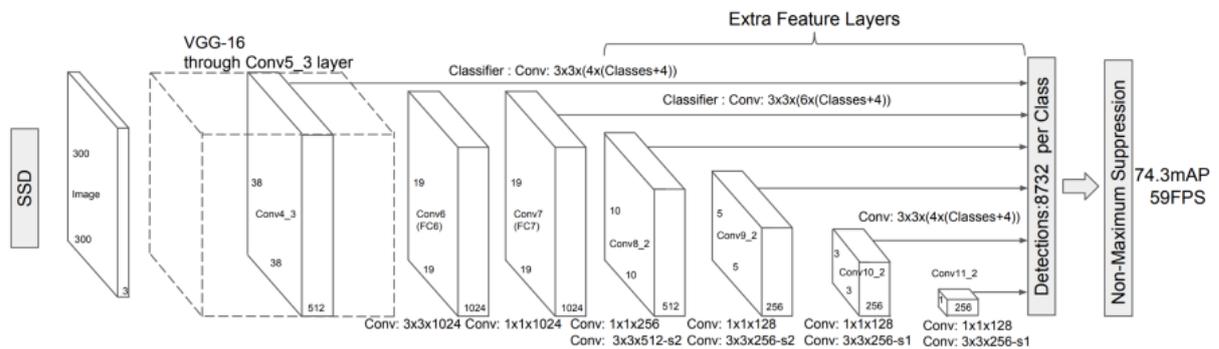


Figura 26 – Arquitetura do método SSD.

Fonte: Liu et al. (2016).

A razão pela qual a VGG-16 foi usada como rede básica é justificada pelo seu forte desempenho em tarefas de classificação de imagens de alta resolução e sua popularidade em problemas que a transferência de aprendizado contribui significativamente com a melhora dos resultados. Conforme pode ser observado na Figura 26, a arquitetura do SSD possui mais seis camadas de convolução auxiliares após a rede VGG-16, em que cinco delas são utilizadas para a detecção de objetos. Os mapas de recursos obtidos em diferentes escalas melhoram significativamente a precisão da detecção.

A ideia principal da etapa de previsão das caixas delimitadoras é parecida com o conceito de âncoras do método *Faster* R-CNN. As caixas propostas são cuidadosamente selecionadas com base em seus tamanhos, proporções e posições na imagem. O objetivo final do modelo é decidir qual caixa delimitadora utilizar em uma determinada imagem e consequentemente prever as compensações que devem ser feitas de modo a ajustar as coordenadas da caixa para obter a previsão final. A Figura 27 retirada do artigo original de Liu et al. (2016) apresenta a ideia da previsão de caixas delimitadoras do método SSD.

Condizente com o ilustrado na Figura 27, após a imagem passar pelas operações de convolução tem-se, para cada local, uma certa quantidade de caixas delimitadoras. Essas caixas têm tamanhos e proporções desiguais, pelo fato de que classes diferentes de objetos possuem formatos distintos. Por exemplo, um retângulo vertical, normalmente, é mais adequado para humanos e um retângulo horizontal é mais pertinente que seja aplicado para carros. Dito isso, a função *MultiBox* consegue medir o grau de confiança da rede quanto à objetividade da caixa delimitadora calculada e, além disso, calcular a que distância as caixas delimitadoras previstas estão das reais, durante o treinamento do

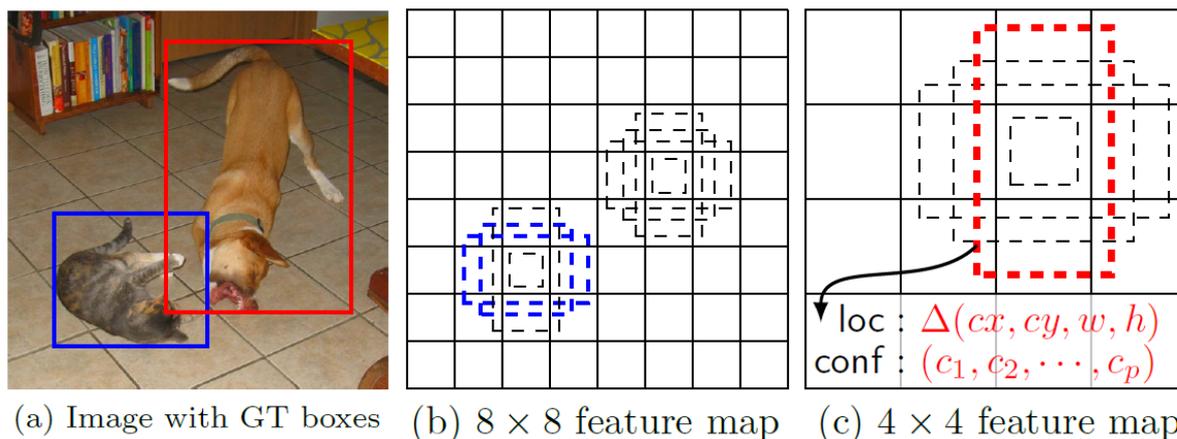


Figura 27 – Funcionamento da previsão de caixas delimitadoras do método SSD.

Fonte: Liu et al. (2016).

modelo.

Nessa etapa de treinamento, o método SSD utiliza alguns recursos que melhoram a eficiência do modelo. Como é de se esperar, a maioria das caixas delimitadoras geradas não enquadram nenhum objeto, ou seja, pertencem a classe negativa. O problema do desbalanceamento entre classes é recorrente em aplicações com imagens, devendo ser tratado de acordo sua especificidade. Com isso, ao invés do SSD usar todos os exemplos negativos, classifica-se a confiança para cada caixa e posteriormente seleciona-se as melhores, de modo que a proporção entre amostras positivas e negativas sejam de no máximo 1:3. A razão pela qual o modelo mantém amostras negativas é que o método também precisa aprender informações que caracterizam uma detecção incorreta. Dessa forma, a aplicação dessa técnica acarreta diretamente em um treinamento mais rápido e estável.

Além da mineração negativa, outra técnica utilizada durante o treinamento do SSD é o aumento de dados. Frequentemente empregado em algoritmos de aprendizado profundo, o aumento de dados é extremamente importante para ensinar a rede a se tornar mais robusta a diferentes tamanhos de objetos e suas posições variadas na imagem. Com isso, são gerados exemplos distintos a partir da imagem original, em diferentes proporções. São realizadas operações de inversão horizontal e distorções fotométricas, garantindo que apesar das alterações na imagem original, as novas imagens geradas mantenham as mesmas características.

Por fim, dado o elevado número de caixas delimitadoras geradas durante a passagem direta da imagem pelo método, o SSD, assim como grande parte dos métodos de detecção

de objetos, aplica a técnica de supressão não máxima. Com isso, caixas delimitadoras que possuem um escore de confiança e um valor de intersecção sobre a união baixo (tais valores são previamente definidos de forma parametrizada) são descartados, sendo preservadas aproximadamente 200 (valor considerado pelos autores) caixas por imagem. Isso garante que apenas as previsões mais prováveis sejam mantidas pela rede, enquanto as não interessantes são removidas.

O método SSD utiliza um maior número de caixas delimitadoras, provocando o aumento da sua acurácia, mas em contrapartida gera um impacto na velocidade. Além disso, o fato de se utilizar o *MultiBox* em várias camadas de convolução resulta em uma melhor detecção, pela razão de que o detector é executado para recursos em diferentes resoluções. Outra constatação observada no trabalho de [Liu et al. \(2016\)](#) é que 80% do tempo de processamento gasto é na rede VGG-16, ou seja, caso se utilize uma rede neural convolucional mais rápida com acurácia parecida (AlexNet, GoogleNet, Resnet, Inception, Xception ou MobileNet), o desempenho do método SSD pode ser ainda mais satisfatório.

O detector de objetos SSD é um método de disparo único, não possuindo uma rede de proposta de região assim como os detectores de objetos da família R-CNN. Ao invés disso, prevê as caixas delimitadoras e suas respectivas classes diretamente dos mapas de recursos obtidos em uma única passagem pela rede. Para ter uma precisão competitiva com os demais concorrentes, o SSD utiliza pequenos filtros de convolução para prever as classes dos objetos e os deslocamentos de ajuste das caixas de precisão. Além disso, assim como feito no método YOLO, o SSD possui âncoras de tamanho padrão para lidar com a diferença de proporção entre objetos e, o seu grande diferencial, que é a utilização dos mapas de recursos em diferentes escalas para realizar a detecção. Sendo assim, pode ser empregado tanto em aplicações que exigem processamento em tempo real quanto para as demais, que demandam uma taxa de acerto satisfatória.

3.5 Resumo

Este capítulo descreveu as principais tarefas na área de visão computacional bem como as principais técnicas de abordagem em cada uma delas. Além disso, o foco do capítulo foi para os métodos de detecção de objetos que atualmente compõe o estado da arte para solucionar tarefas dessa natureza. O próximo capítulo irá apresentar os resultados

obtidos por outros autores sobre os estudos realizados na mesma temática desse trabalho.

Capítulo 4

Trabalhos Relacionados

O capítulo de Trabalhos Relacionados consiste num resumo de discussões já feitas por outros autores sobre os estudos realizados nas áreas de parasitologia e inteligência artificial, servindo como embasamento para o desenvolvimento deste trabalho. Foram consultadas as referências bibliográficas condizentes com a atual linha de interesse e foram verificadas as pesquisas previamente desenvolvidas por alguns autores sobre a mesma temática.

Além disso, também foi descrito sobre os resultados das pesquisas que foram obtidas previamente, indicando os respectivos responsáveis pelas análises, criando um plano de sustentação argumentativo, dando embasamento e servindo como comparação em relação aos resultados a serem obtidos nesta pesquisa.

4.1 Pesquisas

Nas últimas décadas vários autores se dedicaram a estudar formas de automatizar o processo de encontrar e classificar ovos de parasitos em amostras fecais com o auxílio de um microscópio. Grande parte dos trabalhos até o momento se baseia na aplicação de métodos de processamento digital de imagens e reconhecimento de padrões, buscando encontrar características específicas de cada tipo de ovo de parasito para classificá-lo corretamente. Além disso, os autores empregaram técnicas alternativas de preparação das lâminas microscópicas, ao invés de utilizarem o Kato-Katz, o que implica diretamente nos resultados alcançados, uma vez que as lâminas ficam limpas e os ovos muito mais nítidos nas imagens.

Yang et al. (2001) propõem em seu estudo um algoritmo utilizando técnicas de processamento digital de imagens e uma RNA. As características morfométricas de ovos de parasitos humanos em amostras fecais foram extraídas de imagens microscópicas através do processamento digital de imagens. Uma RNA classificou as espécies de parasitos com base nessas características. Os autores selecionaram quatro características morfométricas com base em três características morfológicas que representam forma, suavidade da concha e tamanho. Os resultados obtidos mostraram uma taxa de classificação correta média de 86,1% para a RNA-1 e 90,3% para a RNA-2.

No trabalho de Dogantekin et al. (2008) foi proposta uma técnica baseada em momentos invariantes de uma imagem, que permite calcular a área de um objeto, mesmo que tenha sofrido mudança de tamanho ou tenha sido rotacionado. Foram aplicados alguns métodos de processamento digital de imagens, como redução de ruído, aprimoramento de contraste, segmentação e processamento morfológico, que foram utilizados na etapa de extração de características da abordagem usada neste estudo. A técnica apresentada neste trabalho permite classificar 16 diferentes ovos de parasitos a partir de suas imagens microscópicas. As simulações foram realizadas em computador pessoal no ambiente MATLAB e a taxa geral de acerto variou em torno de 95%.

Assim como Dogantekin et al. (2008), Avci and Varol (2009) também propõem uma metodologia baseada em momentos invariantes, combinando-a com a máquina de vetores de suporte multiclasse (MCSVM). O método proposto contém quatro estágios: pré-processamento, extração de recursos, classificação e teste. Foi utilizado o *software* MATLAB para estimar a taxa de classificação de sucesso da abordagem proposta. Para esse objetivo, o método foi avaliado usando dados do conjunto de teste. No final da avaliação, foi obtido uma taxa geral de acerto de 97,70%.

Delgado Huaynalaya and Hirata (2012) explica que o parasita *S. mansoni* afeta atualmente milhões de pessoas no Brasil, e uma das formas de diagnóstico da esquistossomose é a detecção de ovos do parasita por meio da análise de lâminas microscópicas com material fecal. Esse diagnóstico é difícil e trabalhoso, por isso, os autores propuseram um método que consegue encontrar e extrair características dos ovos do *S. mansoni* para uma posterior detecção. Foram utilizados operadores morfológicos visando realçar os contornos e o formato elíptico dos ovos. Os resultados experimentais mostram que o método é capaz de detectar entre 75% a 82% dos ovos em que os contornos são visíveis, ou seja, que estão

evidentes em lâminas que não possuem impurezas ou sujeiras em abundância.

No estudo de [Suzuki et al. \(2012\)](#) foi proposto um método capaz de segmentar e classificar, a partir de imagens de microscopia com impurezas fecais, as 15 espécies mais comuns de cistos de protozoários, ovos de helmintos e larvas no Brasil. A abordagem desenvolvida explora o formato dos objetos elipsoidais e a transformada imagem-floresta para segmentação de imagens. Os resultados com uma revocação 90,38% e precisão de 98,32% indicam que o método implementado é uma abordagem promissora para a automação completa do diagnóstico de enteroparasitoses.

[Suzuki et al. \(2013\)](#) mencionam que, por mais que existam alguns métodos de identificação automática de ovos de parasitos, ainda existem algumas espécies que não obtiveram um índice aceitável de acerto, por isso, os autores propõem uma solução que visa automatizar o diagnóstico das 15 espécies mais comuns de enteroparasitos no Brasil, usando uma técnica parasitológica sensível, um microscópio motorizado com câmera digital para aquisição e foco automático de imagens e métodos de análise de imagens. Os resultados de revocação de 93% e precisão de 99,17% indicam que a solução proposta é eficaz e adequada para a rotina de laboratório, na qual o exame passa a ser concluído em alguns minutos.

[Nkamgang et al. \(2018\)](#) apresentam um sistema computacional que analisa automaticamente imagens microscópicas e consegue classificar parasitos intestinais. Para treinar o algoritmo, os autores reuniram imagens de diferentes sites ^{1 2 3}. A abordagem é baseada na segmentação e treinamento de um classificador, utilizando a combinação de um sistema neuro-fuzzy e uma RNA. O parasito é primeiro localizado pela transformação circular de Hough e depois extraído para segmentação. Os resultados demonstram uma classificação com uma taxa de reconhecimento de 100% para cada uma das 20 classes de parasitos abordadas na pesquisa.

No estudo de [Nkamgang et al. \(2019\)](#) foi implementado um sistema médico especialista automatizado e útil para o diagnóstico da parasitose intestinal humana para 20 espécies. O *software* foi desenvolvido com base em um algoritmo de decisão. Uma base de conhecimento foi construída através de informações obtidas de livros e médicos, com

¹ http://bioimage.free.fr/par_image/parasites_aliments.htm

² https://apps.who.int/iris/bitstream/handle/10665/69987/WHO_CTD_SIP_98.2_Slides_1-45.pdf

³ https://apps.who.int/iris/bitstream/handle/10665/69987/WHO_CTD_SIP_98.2_Slides_46-90.pdf

informações referentes a cada doença causada por um parasito. O usuário interage com o sistema, respondendo a perguntas. As informações obtidas são cruzadas com um segundo método que utiliza a transformada circular de Hough e um classificador neuro-fuzzy treinado. O sistema foi avaliado com 60 casos de infecção e comparado ao diagnóstico de dois médicos especialistas. Dos 60, foram obtidos 58 diagnósticos corretos, o que corresponde a uma sensibilidade de 96,6%.

Tchinda et al. (2019) ofereceram uma solução para automatizar o diagnóstico de parasitos intestinais através de imagens obtidas de um microscópio conectado diretamente a um computador. A abordagem explora a detecção de contorno com base na transformação de *wavelets* em várias escalas para detectar o parasito. Os contornos ativos são combinados com a transformação Hough para realizar a segmentação e extração da imagem do parasito. A ferramenta de classificação é baseada em uma rede neural probabilística. O algoritmo desenvolvido foi testado em 900 amostras de imagens microscópicas de 15 espécies diferentes de parasitos intestinais. O resultado para o teste realizado foi o reconhecimento correto de 100% das amostras avaliadas.

Na pesquisa de Viet et al. (2019) foi utilizado o método de detecção de objetos *Faster* R-CNN para localizar ovos de parasitos intestinais humanos em imagens com material fecal. Os autores executaram alguns experimentos variando o conjunto de dados abordado. Para avaliar os resultados, as métricas utilizadas foram a precisão e a revocação. Os experimentos realizados obtiveram excelentes resultados, alcançando um AP de 97,67% consolidando a utilização do método *Faster* R-CNN na detecção de ovos de parasitos intestinais humanos. O conjunto de dados utilizado pelos autores é disponibilizado em (<http://www.k-state.edu/parasitology/>) pela Universidade Kansas State. A Figura 28 apresenta algumas imagens de ovos da espécie *S. mansoni* do *dataset*.

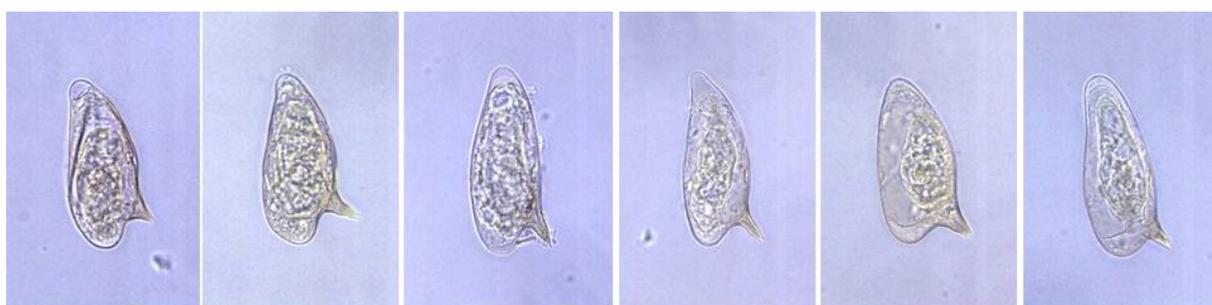


Figura 28 – Imagens de *S. mansoni* disponibilizadas pela Universidade Kansas State.

Foi discutido no trabalho de Peñas et al. (2020) a importância de levar em

consideração o quão grave pode ser os sintomas de pessoas infectadas por algumas espécies de parasitos, dentre elas a *S. mansoni*. Essas doenças podem inclusive causar a morte, o que acontece frequentemente principalmente em países pobres. Sendo assim, os autores empregaram o método de detecção de objetos YOLO para encontrar ovos de helmintos em amostras de fezes. Foi coletado um conjunto de imagens em diferentes condições de iluminação e variação da câmera. Nos experimentos realizados os autores alcançaram uma alta sensibilidade na detecção de ovos de helmintos e uma alta acurácia na identificação das espécies, obtendo um valor de AP de 86,83%. Portanto, pode-se concluir que faz sentido a utilização do modelo treinado para o diagnóstico automatizado de ovos de parasitos.

No estudo de Larsson and Hedberg (2020) foi realizado um comparativo entre técnicas de aprendizado profundo para o problema de classificação de ovos de parasitos intestinais humanos. A comparação foi entre três métodos de detecção de objetos variando algumas arquiteturas de CNN, foram eles: SSD MobileNetV2, SSD Inception, *Faster R-CNN ResNet101*, *Faster R-CNN ResNet50* e *R-FCN ResNet101*. As espécies de parasitos abordadas foram a *Ascaris lumbricoides*, *Trichuris trichuria*, *Hookworm* e *Schistosoma mansoni*. De acordo com os autores, o melhor método testado foi o *R-FCN ResNet101*, em que obteve um valor médio de acurácia (94,25%) maior que os demais, mesmo possuindo algumas limitações, principalmente em relação aos falsos negativos. O estudo se mostrou promissor para pesquisas futuras, encorajando demais pesquisadores para abordarem o tema.

Além do *S. mansoni*, outro helminto frequentemente estudado se trata do *A. lumbricoides*. Na pesquisa de Butploy et al. (2021) foi empregada uma CNN na tarefa de classificação de ovos da *A. lumbricoides* para o diagnóstico da doença. O conjunto de imagens foi preparado usando uma técnica de concentração de acetato de etila formalina, realçando o ovo no meio das sujeiras, facilitando para o especialista durante sua inspeção visual. Como resultados, os autores obtiveram uma acurácia de 93,33% nas imagens de teste.

No estudo de Dacal et al. (2021) os autores se dedicaram em desenvolver uma plataforma para auxiliar os especialistas no diagnóstico de doenças causadas pelas espécies Ancilostomídeos, *A. lumbricoides* e *T. trichiura*. Enfatizando que a técnica parasitológica de preparação de lâminas recomendada pela OMS é a Kato-Katz, os autores utilizaram da mesma para a realização do trabalho. Foi empregado o método de detecção de objetos

SSD com uma rede MobileNet, inicializada com pesos pré-treinados no *dataset* COCO. O modelo foi treinado e testado em 41 lâminas de amostras de fezes, obtendo uma precisão média de 98,44% e revocação média de 80,94% nos experimentos realizados.

Uma série de trabalhos (Sa et al., 2017; Ding et al., 2017; Yang et al., 2018; Akselrod-Ballin et al., 2019; Rosati et al., 2020; Mahmood et al., 2020; Sheng et al., 2020) já utilizaram o método de detecção de objetos *Faster* R-CNN no campo de imagens médicas. A técnica ao ser comparada com outros detectores do estado da arte, como YOLO e SSD, demonstrou possuir uma maior precisão em detectar objetos, apesar de possuir um tempo de inferência maior. No problema abordado nesse estudo, a maior preocupação é em relação a detecção correta dos ovos, e não necessariamente com o tempo de inferência que a solução gasta para realizar a detecção.

Sendo assim, um dos objetivos específicos dessa pesquisa foi fazer o uso *Faster* R-CNN para automatizar o processo de diagnóstico de ovos de parasitos da espécie *S. mansoni*, treinando, avaliando e testando diferentes arquiteturas de CNN para fazer parte do método. Além disso, abordar outros métodos de detecção de objetos e suas diferentes configurações de arquitetura que vão além do *Faster* R-CNN, tais como o SSD e o YOLO.

Na Tabela 3, são resumidas as principais informações dos artigos revisados que utilizaram inteligência artificial para classificar ovos de parasitos humanos em imagens microscópicas. É apresentada uma breve descrição dos artigos, sendo composta de como foram coletadas as imagens e conseqüentemente a técnica de preparação das lâminas, os métodos usados e os resultados para as métricas de desempenho informadas.

Tabela 3 – Resumo dos trabalhos relacionados.

Referência	Conjunto de imagens	Técnica parasitológica	Métodos	Métricas avaliadas
Yang et al. (2001)	Coleta própria	Não informada	Processamento Digital de Imagens e MLP Classifier	Acurácia: 90,3%
Dogantekin et al. (2008)	Universidade Kansas State	Não informada	Momentos invariantes e sistema de inferência fuzzy baseado em rede adaptativa (IMANFIS)	Acurácia: 95%
Avcı and Varol (2009)	Universidade Kansas State	Não informada	Momentos invariantes e SVM multiclasse (MCSVM)	Acurácia: 97,7%
Delgado Huaynalaya and Hirata (2012)	Coleta própria	Kato-Katz	Operadores morfológicos e transformada de Hough	Revocação: 78,5%
Suzuki et al. (2012)	Coleta própria	TF-Test	Segmentação e OPF	Revocação: 90,38% Precisão: 98,32%
Suzuki et al. (2013)	Coleta própria	TF-Test Modificado	Segmentação e OPF	Revocação: 93% Precisão: 99,17%
Nkamgang et al. (2018)	Reunidos de diferentes sites	Não informada	Transformada circular de Hough e classificador neuro-fuzzy	Acurácia: 100%
Nkamgang et al. (2019)	Reunidos de diferentes sites	Não informada	Transformada circular de Hough e classificador neuro-fuzzy combinado com sistema especialista	Acurácia: 96,6%
Tchinda et al. (2019)	Coleta própria	Não informada	Transformada wavelet multi-escala, transformada de Hough e rede neural probabilística	Acurácia: 100%
Viet et al. (2019)	Universidade Kansas State	Não informada	<i>Faster</i> R-CNN	AP: 97,67%
Peñas et al. (2020)	Coleta própria	Não informada	YOLO	AP: 86,83%
Larsson and Hedberg (2020)	Coleta própria	Não informada	R-FCN ResNet101	Acurácia: 94,25%
Butploy et al. (2021)	Coleta própria	formalina-acetato de etila	CNN para classificação	Acurácia: 93,33%
Dacal et al. (2021)	Coleta própria	Kato-Katz	SSD com MobileNet	Revocação: 80,94% Precisão: 98,44%

É possível perceber nos trabalhos relacionados que, como grande parte dos autores não são da área médica, houve um descuido de não informar corretamente sobre como foi feita a coleta dos dados, além de não deixarem claro qual foi a técnica de preparação das lâminas que continham amostra fecal dos pacientes. Outro grande problema notado foi que a maioria dos trabalhos não consideraram a métrica de avaliação IoU, a qual é extremamente importante em tarefas de detecção de objetos, uma vez que com seu uso é possível comparar exatamente o local onde foi gerada a caixa delimitadora em torno do objeto, e onde realmente ela deveria ter sido gerada.

Dessa forma, considerando os trabalhos encontrados na literatura, entendeu-se a importância e o desenvolvimento de cada um. Entretanto, concluiu-se que nenhuma pesquisa relacionada aparentemente tenha sido desenvolvida com um enfoque de aplicação prática em ambiente real, dado as limitações apresentadas pelas soluções, em especial por não usarem a técnica de preparação de lâminas Kato-Katz.

Além disso, sabe-se que é uma opção válida e promissora a tentativa em generalizar modelos de aprendizado profundo para várias classes, porém, tal abordagem possui um custo que interfere diretamente nos resultados. Sendo assim, diferentemente dos trabalhos relacionados que almejavam o diagnóstico para diferentes espécies de parasitos intestinais humanos, na solução aqui apresentada se buscou um método especializado no diagnóstico da esquistossomose.

Isto posto, uma importante contribuição da pesquisa foi a aquisição de um conjunto de imagens coletadas em ambiente real, com o auxílio de especialistas da área de parasitologia. Além disso, o uso das lâminas preparadas por meio da técnica Kato-Katz, diferindo dos demais trabalhos relacionados, que empregaram técnicas alternativas. Por fim, o treinamento e o teste de diferentes técnicas de inteligência artificial de última geração, com o objetivo de a solução proposta ser colocada em uso e auxiliar o especialista em sua tomada de decisão, para o diagnóstico da esquistossomose.

4.2 Resumo

Neste capítulo foram discutidos os resultados obtidos por outros autores sobre os estudos realizados na mesma temática desse trabalho. Foi realizada uma análise comparativa resumida entre as pesquisas e apresentada de forma tabular, facilitando o entendimento

das principais informações de cada estudo. Ademais, o próximo capítulo tem como objetivo descrever a metodologia dessa pesquisa, citando o material e ferramentas utilizadas, além de apresentar todas as etapas metodológicas que foram aplicadas no decorrer de seu desenvolvimento.

Capítulo 5

Material e Métodos

O capítulo de Material e Métodos descreve o planejamento detalhado sobre o que foi realizado nessa pesquisa, de forma que o trabalho possa ser executado por outros pesquisadores, para a replicabilidade. Todas as etapas do processo foram descritas, desde a aquisição do conjunto de imagens até a validação dos modelos propostos através de experimentos e comparação de resultados.

Cada etapa do desenvolvimento dessa pesquisa pode ser resumida no diagrama de blocos apresentado na Figura 29 e detalhada na lista a seguir.

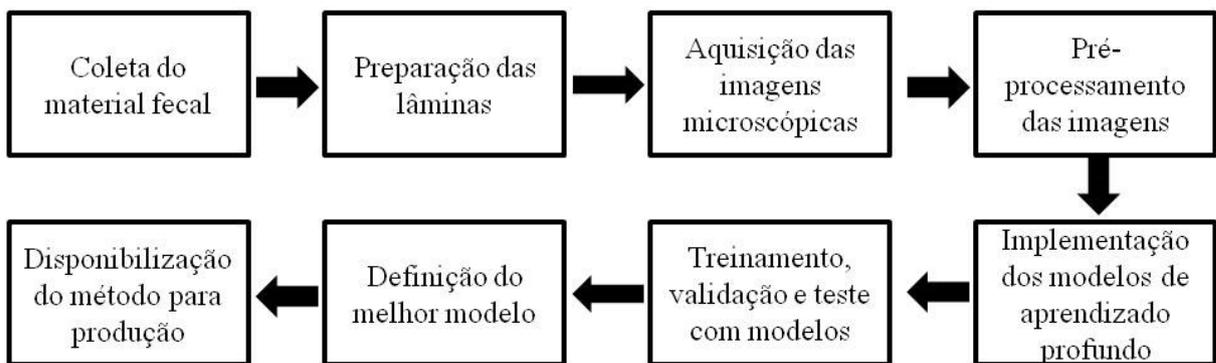


Figura 29 – Diagrama de blocos ilustrando as etapas de desenvolvimento do estudo.

1. Coleta do material fecal de diferentes pacientes.
2. Preparação das lâminas pela técnica Kato-Katz.
3. Aquisição das imagens através de diferentes modelos de microscópios.
4. Preparação do conjunto de dados e pré-processamento das imagens obtidas.

5. Implementação, treinamento e testes com os métodos de detecção de objetos baseados em aprendizado profundo abordados na pesquisa.
6. Implementação, treinamento e testes com os modelos de aprendizado de máquina empregados para integrar a solução final.
7. Análise para definição da melhor configuração para o sistema proposto e discussão dos resultados.
8. Disponibilização da ferramenta desenvolvida em ambiente de produção através de uma aplicação *Web*.

5.1 Coleta e pré-processamento do conjunto de imagens

Com o auxílio do Laboratório de Imunologia de Helmitos no Instituto de Ciências Biológicas da Universidade Federal de Minas Gerais, preparou-se as lâminas pelo método Kato-Katz contendo amostras fecais de diferentes pacientes. Em seguida, as lâminas foram levadas ao microscópio para serem examinadas. Foram utilizados dois microscópios clínicos, o primeiro da marca Olympus, modelo BX41, ajustado para um *zoom* de 100 vezes, tendo sido capturadas 573 imagens em RGB com resolução de 2048×1536 . Destas, 530 imagens continham pelo menos um ovo da espécie *S. mansoni*, enquanto o restante, 43 imagens, continham sujeiras, impurezas e ovos de outras quatro espécies de parasitos intestinais humanos, sendo eles: *Ancylostoma duodenale*, *Necator americanus*, *Ascaris lumbricoides*, *Enterobius vermicularis* e *Trichuris trichiura*.

O segundo microscópio utilizado é da marca *Scientific Instrument & Optical Sales*, modelo EX30, ajustado para um *zoom* de 100 vezes, tendo sido capturadas 527 imagens em RGB com resolução de 2688×1520 . Destas, todas as imagens continham pelo menos um ovo da espécie *S. mansoni*. A quantidade de imagens obtidas é justificada pelas amostras disponíveis no momento da coleta, enquanto os valores para o *zoom* e a resolução das imagens foram definidos de acordo com o que se é comumente utilizado nesse tipo de análise realizada pelos especialistas em laboratório.

Após a coleta das imagens, dois especialistas treinados da área de parasitologia, fizeram o uso da ferramenta gráfica de anotação de imagens e rótulos de caixas delimi-

tadoras de objetos em imagens LabelImg (Tzutalin, 2015) (Figura 30). Cada um dos especialistas analisou cada uma das 1100 imagens coletadas individualmente, não havendo um enviesamento por parte de um ou de outro. Dessa forma, foram marcados cada um dos ovos da espécie *S. mansoni* com suas respectivas coordenadas (formato Pascal VOC) de onde se encontram na imagem. Sendo assim, cada uma das 1057 imagens que possuíam um ovo da espécie *S. mansoni* foram anotadas e conseqüentemente gerados 1057 arquivos XML, que armazenam as informações acerca das imagens originais, incluindo as coordenadas dos ovos em cada uma dessas imagens.

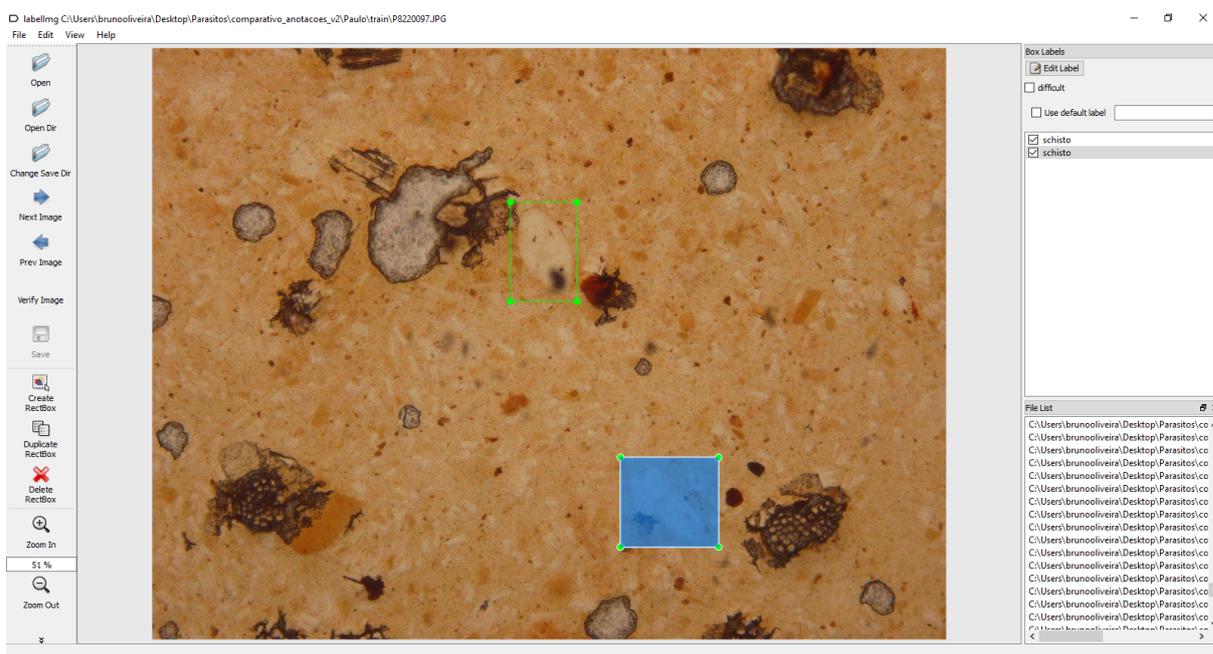


Figura 30 – Exemplo de uso do LabelImg pelo especialista.

As demais 43 imagens que não continham ovos da espécie abordada também foram analisadas pelos dois especialistas que confirmaram não existir nenhum ovo da espécie *S. mansoni*. Tais imagens serão utilizadas exclusivamente no conjunto de teste, com o objetivo de verificar a incidência de falsos positivos, uma vez que não contém ovos da espécie abordada. Para cada uma dessas 43 imagens também foram gerados 43 arquivos XML, que armazenam informações como as dimensões e nome da imagem, porém, nenhuma marcação de objeto, uma vez que não foram geradas caixas delimitadoras.

Dessa forma, como cada uma das 1100 imagens foram anotadas pelos dois especialistas, teve-se ao final do processo 1100 arquivos XML que correspondem às anotações realizadas pelo especialista A e 1100 arquivos XML que correspondem às anotações realizadas pelo especialista B.

Em seguida, foi feita uma análise criteriosa entre as anotações realizadas pelos especialistas, tendo sido encontradas algumas divergências entre as anotações, o que colabora com a justificativa de quanto o problema do diagnóstico da esquistossomose por meio de imagens microscópicas pode ser difícil, até mesmo para profissionais experientes. De todas as 1100 imagens, o especialista A encontrou 1186 ovos da espécie *S. mansoni*, enquanto o especialista B encontrou 1210. A quantidade de imagens que divergiu entre os dois especialistas totaliza 31, somadas com as 43 imagens que não possuem nenhum ovo da espécie *S. mansoni*, representam o conjunto de teste. A Figura 31 apresenta seis exemplos de imagens que causaram divergência entre as opiniões dos especialistas. Para cada imagem, à esquerda é ilustrada as caixas delimitadoras desenhadas pelo especialista A e à direita as desenhadas pelo especialista B, para cada uma das seis imagens do exemplo.

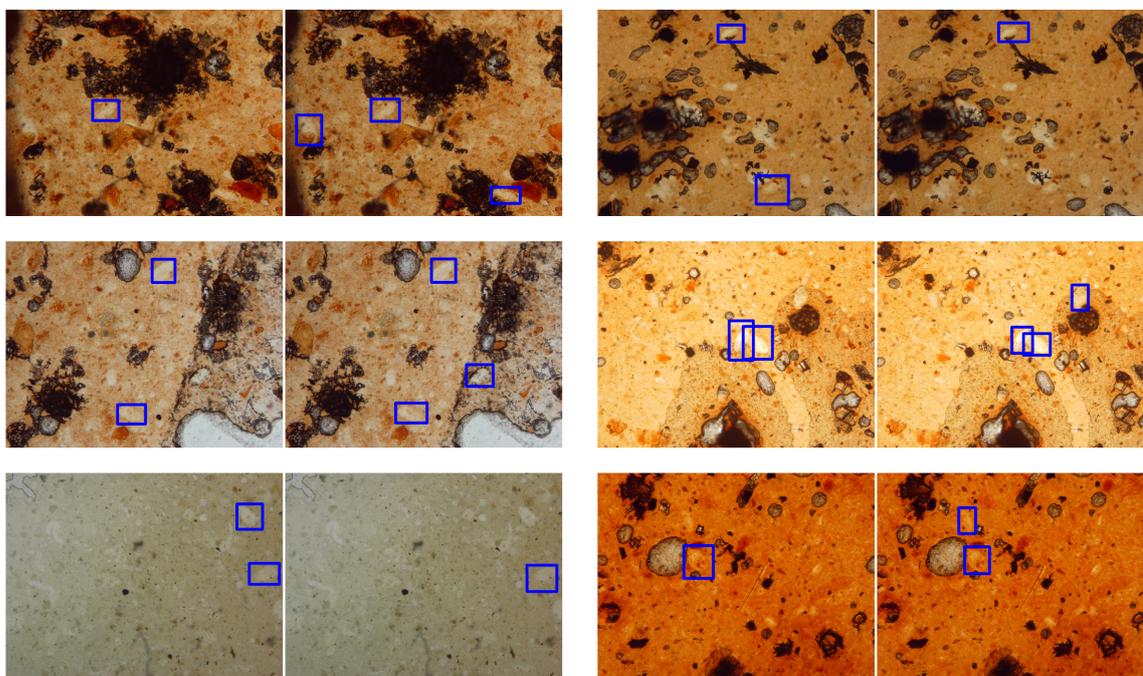


Figura 31 – Exemplos de imagens em que as anotações dos especialistas divergiram.

Como houve divergência entre os especialistas em algumas imagens, não seria possível obter um conjunto de teste confiável para a avaliação dos modelos. Dessa forma, um terceiro especialista em parasitologia foi convidado para anotar as imagens contidas no conjunto de teste, que totalizam 74. Com as marcações dos três especialistas, foi possível produzir um conjunto de teste sendo considerado como ovos da espécie *S. mansoni* somente aqueles que foram encontrados por no mínimo dois especialistas.

Os estudos de [Beam et al. \(2020\)](#) e [Uçar et al. \(2020\)](#) discutem os desafios e os problemas de se aplicar a aleatoriedade nos dados para o treinamento e teste dos algoritmos

de aprendizado, em especial na área da saúde. Dessa forma, a escolha por usar as imagens divergentes no conjunto de teste é justificada pelo fato de aumentar a capacidade de reprodutibilidade dos experimentos, além de testar a solução com um conjunto de imagens consideradas difíceis, uma vez que até mesmo especialistas treinados tiveram opiniões divergentes acerca dessas imagens.

Pelo fato de as marcações de todas as imagens terem sido anotadas por dois especialistas diferentes, foi necessário definir critérios para selecionar qual conjunto de treinamento utilizar, o do especialista A ou do especialista B. Para cada caixa delimitadora que envolve o ovo da espécie *S. mansoni*, foi calculada a área em *pixels*. A Figura 32 ilustra o gráfico das áreas calculadas das marcações realizadas pelo especialista A. Para o especialista B, os mesmos critérios foram adotados. A Figura 33 apresenta o gráfico das áreas calculadas das marcações realizadas pelo especialista B.

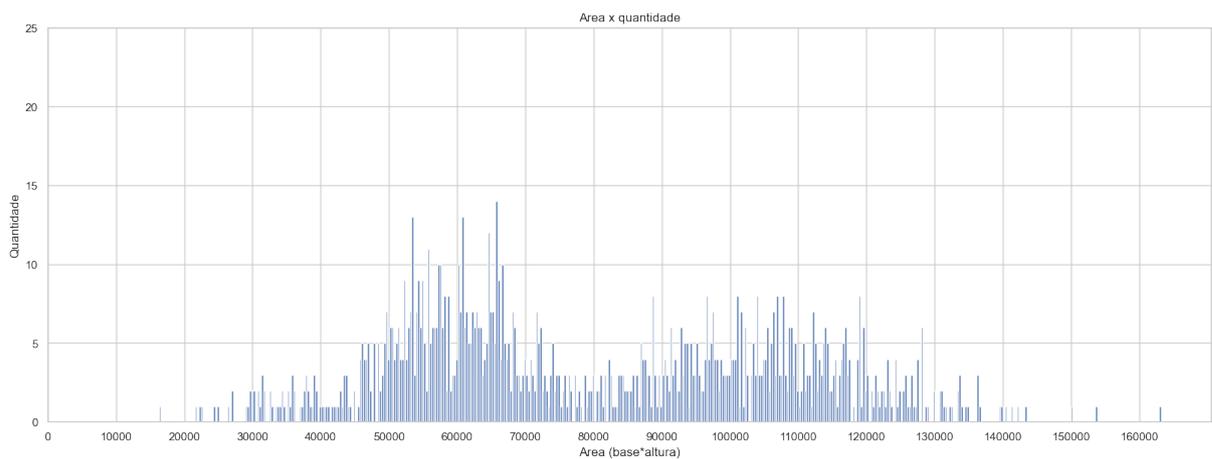


Figura 32 – Distribuição das áreas das caixas delimitadoras do especialista A.

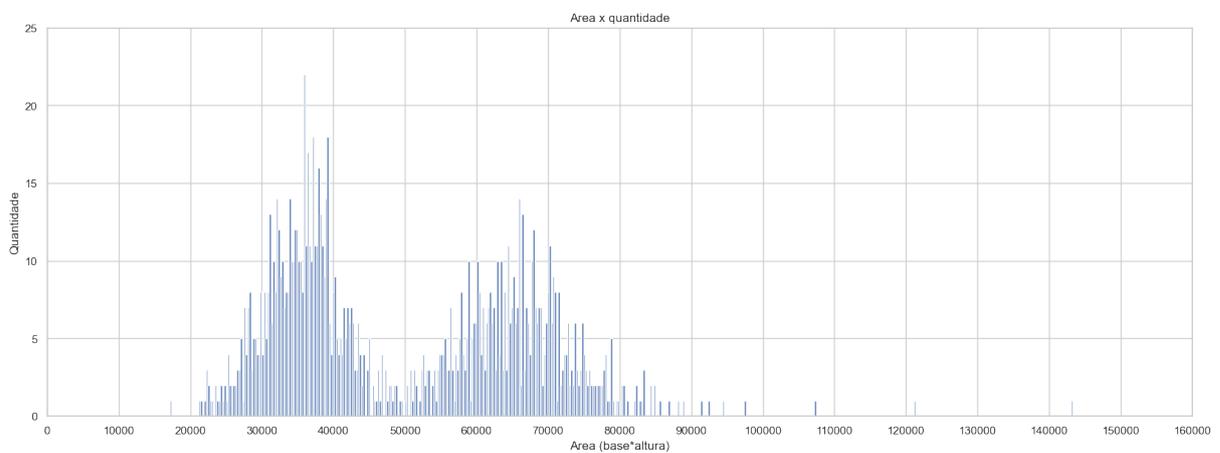


Figura 33 – Distribuição das áreas das caixas delimitadoras do especialista B.

Como pode ser observado, as caldas da distribuição apresentam valores distantes, o que não poderia, uma vez que o tamanho do ovo da espécie *S. mansoni* é padronizado ($110\ \mu\text{m} \times 40\ \mu\text{m}$) (Roberts et al., 2006), considerando a mesma configuração do microscópio. Além disso, com o gráfico mostrado na Figura 33, é possível perceber uma distribuição menos espalhada, ao ser comparada com a distribuição do especialista A apresentada na Figura 32.

A Figura 34 mostra o *boxplot* dos valores que representam o cálculo da área para cada caixa delimitadora anotada pelo especialista A e pelo Especialista B.

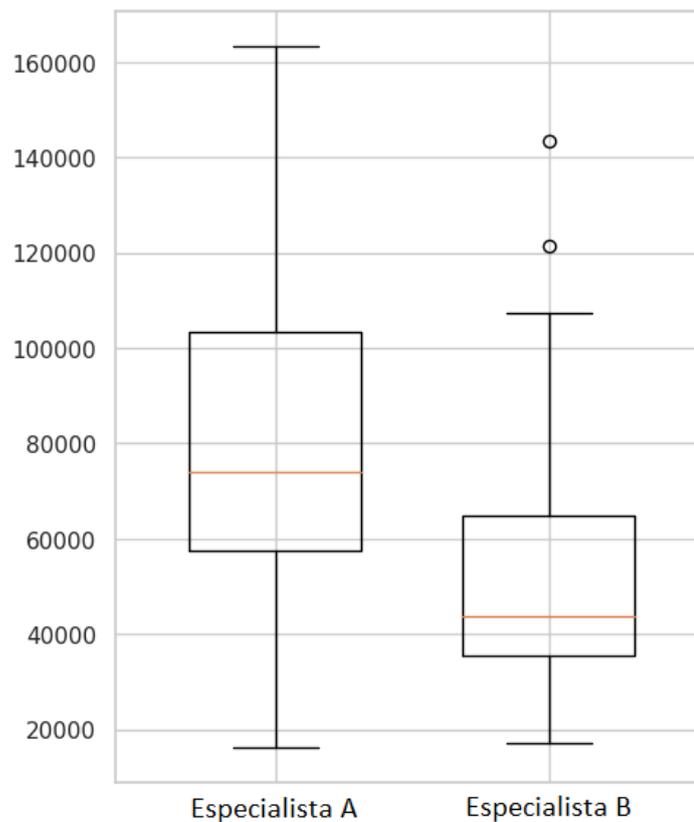


Figura 34 – Boxplot das áreas das caixas delimitadoras dos especialistas A e B.

Na Figura 34 no *boxplot* do especialista A é possível perceber a distância considerável entre o valor mínimo e o valor máximo da distribuição. Além disso, foi calculado o desvio padrão de toda amostra gerada, resultando em um valor de 27249 *pixels* quadrados para a área das caixas delimitadoras anotadas pelo especialista A.

Como ilustrado, o *boxplot* do especialista B da Figura 34 possui alguns poucos valores atípicos, ou seja, observações que apresentam um afastamento das demais da distribuição. Apesar disso, é notório o menor desvio em relação à média das anotações

realizadas pelo especialista B. Para comprovar isso, foi calculado o desvio padrão que resultou em um valor de 16963 *pixels* quadrados, bem inferior ao desvio padrão calculado da distribuição gerada pelas áreas das caixas delimitadoras anotadas pelo especialista A.

Além disso, dois diferentes modelos de aprendizado profundo para a detecção de objetos foram treinados isoladamente, porém, utilizando o mesmo conjunto de hiperparâmetros para uma comparação justa. O objetivo foi treinar os modelos com o conjunto de treinamento do especialista A e outro diferente com as imagens de treino anotadas pelo especialista B, sendo que ambos serão avaliados no mesmo conjunto de teste. Com isso, foi possível definir qual seria as marcações a serem utilizadas no conjunto de treino, sendo do especialista A ou as do especialista B.

Sendo assim, o conjunto de treinamento é composto 1026 imagens que contém ovos da espécie abordada, enquanto o conjunto de teste possui 74 imagens. Essa metodologia é justificada uma vez que se buscou selecionar as imagens consideradas difíceis (que houve divergência entre os especialistas treinados) para testar a solução proposta. Além disso, empregar todas as 43 imagens contendo somente sujeiras e impurezas para o conjunto de teste, com o objetivo de verificar se a ferramenta desenvolvida não iria encontrar ovos da espécie *S. mansoni* em imagens que não possuem tais elementos.

Importante ressaltar que a maior dificuldade dos especialistas que varrem essas amostras fecais em busca de classificar os ovos não está em saber identificar corretamente as diferentes espécies de helmintos, mas sim em confundir determinada morfologia de um ovo com alguma sujeira ou impureza contida na amostra fecal. Por isso, ainda durante a coleta das imagens, teve-se a preocupação de se conseguir lâminas que possuíssem sujeiras e impurezas em abundância.

Conforme descrito pela literatura, técnicas de aprendizado profundo exigem um conjunto com milhares de imagens para treinamento. Obtidas as imagens originais e sabendo que apenas 1026 amostras podem acarretar em um desempenho que não faça sentido a utilização das técnicas de aprendizado profundo, optou-se por empregar operações de aumento de dados nessas imagens adquiridas. Ao se aplicar tais operações, é possível elevar a quantidade de imagens consideravelmente.

Dito isso, para cada imagem no conjunto de treinamento, foi aplicada a operação de rotação com o objetivo de realizar o aumento de dados nas imagens. Existem dois

tipos de aumento de dados: *offline* e *online*. O aumento de dados *offline* é quando se cria imagens antes de começar a etapa de treinamento, de forma que essa abordagem é totalmente controlada pelo programador. O aumento de dados *online* é realizado durante o treinamento do modelo, em que para cada época, cria-se imagens alteradas a partir das originais. Dessa forma, para o aumento *offline*, foi feito um aumento com fator oito para cada imagem, totalizando 8208 imagens com ovos da espécie *S. mansoni*.

Cada abordagem de aumento de dados possui suas vantagens e desvantagens. Por exemplo, no aumento *offline* é possível reproduzir exatamente o mesmo conjunto de dados utilizado em diferentes modelos, enquanto no aumento *online* não, facilitando a reprodutibilidade dos experimentos ao se adotar o aumento *offline*. Além disso, operações de aumento de dados são realizadas pelas CPUs, logo, as GPUs devem ficar aguardando as CPUs concluírem o processo durante o treinamento. Em contrapartida, o aumento *online* proporciona uma maior variabilidade do conjunto de treinamento, aumentando a capacidade de generalização do modelo. Neste trabalho foram utilizadas as duas abordagens. Para o aumento *offline*, aplicou-se a operação de rotação em 90, 180 e 270 graus, além da inversão horizontal nas imagens originais e rotacionadas. No aumento *online*, aplicou-se diversas operações que serão citadas no decorrer do texto.

Ao se tratar especificamente do problema abordado nessa pesquisa, sabe-se que os ovos da espécie *S. mansoni* podem ser encontrados em qualquer posição ou disposição na imagem. Dessa forma, optou-se por empregar nas imagens do conjunto de teste as mesmas operações realizadas no conjunto de treinamento para o aumento *offline*. Sendo assim, das 74 imagens do conjunto de teste, após um aumento com fator 8, totalizou-se uma quantidade de 592 imagens. Ressalta-se que, antes da utilização das técnicas de aumento de dados, as imagens já haviam sido separadas para participarem ou do conjunto de treinamento ou do conjunto de teste, não sendo possível que uma mesma imagem, mesmo após a aplicação de alguma operação de aumento de dados, participasse de ambos os conjuntos. A utilização das técnicas de aumento de dados nas imagens de teste foi exclusivamente com o objetivo de verificar se a solução proposta conseguiria identificar o ovo da espécie abordada, mesmo ele tendo sido rotacionado. A Figura 35 ilustra a aplicação das operações de aumento de dados *offline*, mostrando imagem original e suas diferentes versões criadas artificialmente.

Na coluna à esquerda da Figura 35 é apresentada na primeira linha a imagem

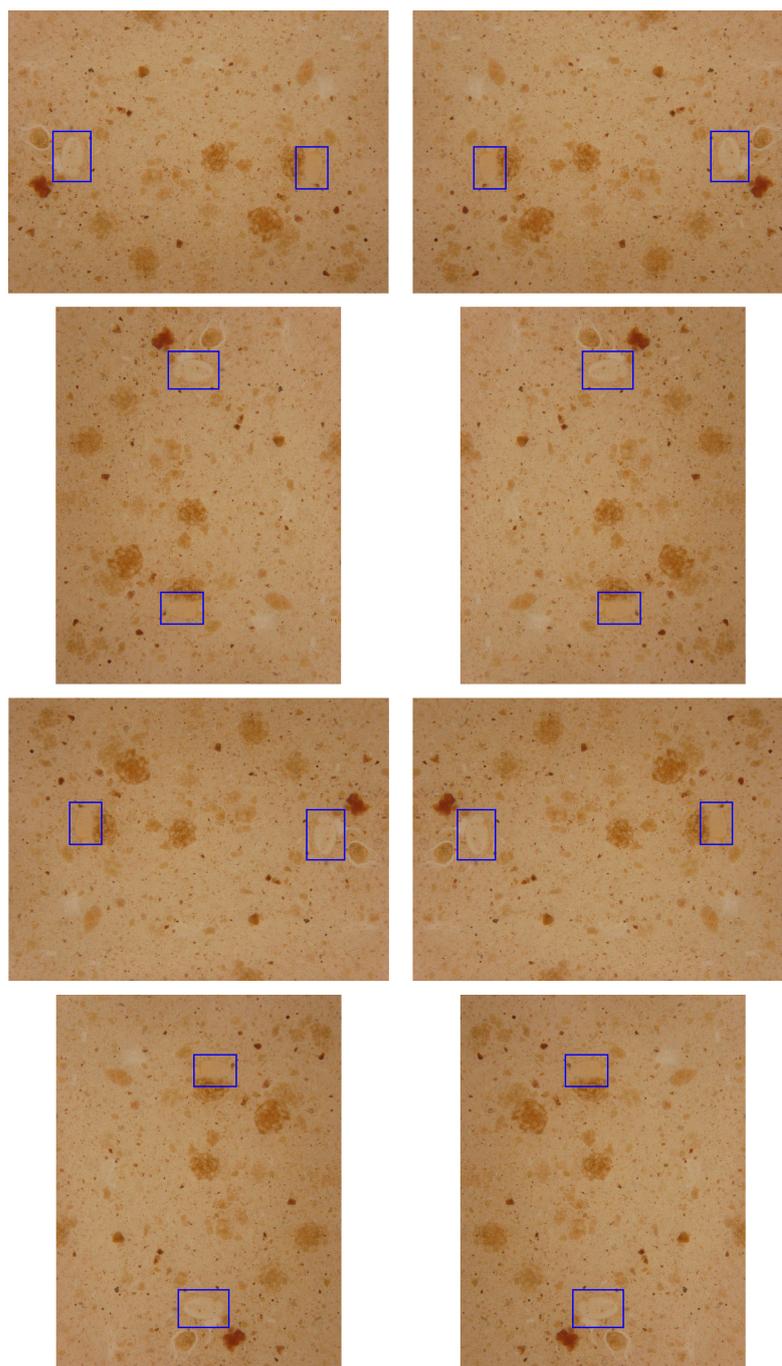


Figura 35 – Aplicação do aumento de dados *offline*.

original sem nenhuma alteração, seguida pela imagem original rotacionada em 90, 180 e 270 graus, respectivamente na segunda, terceira e quarta linha. Já na coluna à direita da Figura 35 é ilustrada as mesmas imagens da coluna à esquerda, após a aplicação da operação de inversão horizontal.

5.2 Definição dos métodos de detecção de objetos

Após a coleta e preparação dos dados, foi definido o material a ser empregado no desenvolvimento do trabalho. O computador utilizado para a implementação dos códigos e realização dos experimentos foi um ASUSTeK, modelo PRIME H310M-E R2.0/BR. O sistema operacional instalado foi o Ubuntu Desktop 18.04.5 LTS, processador Intel(R) Core(TM) i7-9700KF CPU @ 3.60GHz, memória ram 33GB, disco rígido 1TB. Além disso, uma placa de vídeo da marca NVIDIA, modelo GeForce RTX 2080, com arquitetura Turing, Boost Clock 1800 MHz (OC), memória 8GB de GDDR6 e velocidade 14Gbps.

A linguagem de programação empregada foi a Python 3 (Van Rossum and Drake, 2009) e os códigos foram implementados no ambiente de desenvolvimento Jupyter Notebook (Kluyver et al., 2016). Foi utilizada a plataforma de código aberto TensorFlow (Abadi et al., 2016), devido sua flexibilidade de ferramentas, bibliotecas e recursos disponíveis para facilitar o desenvolvimento de aplicações baseadas em aprendizado de máquina. Para os métodos de detecção de objetos *Faster R-CNN* e *SSD* foi empregada a API *Object Detection* (Huang et al., 2017b) e para o método *YOLOv3* a biblioteca *ImageAI* (Moses and Olafewa, 2018). Além disso, foram instalados o *CUDA Toolkit 10.1* e o *CuDNN 7.6.5*, para que fosse possível fazer uso dos recursos da GPU.

O TensorFlow oferece uma coleção de modelos de detecção de objetos pré-treinados no conjunto de dados *COCO* (Lin et al., 2014), que é um conjunto de imagens com 91 classes e é considerado um dos principais *benchmark* na área de visão computacional e aprendizado profundo. Apesar dos modelos pré-treinados conseguirem detectar 91 classes, ovos de parasitos da espécie *S. mansoni* não estão inclusos, fazendo sentido retrainar os modelos. Existem algumas vantagens em se retrainar um modelo pré-treinado em comparação de se treinar um modelo do zero, são elas: o modelo tende a convergir mais rapidamente, não é necessário tantos dados rotulados e os pesos já foram ajustados para um determinado problema em um longo período de treinamento e em computadores com um *hardware* poderoso.

De um modo geral um método de detecção de objetos possui duas partes principais: uma arquitetura de CNN e uma metodologia de se propor regiões candidatas. Existem inúmeras arquiteturas de CNN já consolidadas no estado da arte de visão computacional e cada uma possui diferentes topologias e apresenta vantagens e desvantagens. Algumas delas

são: AlexNet (Krizhevsky et al., 2017); DarkNet (Redmon, 2016); DenseNet (Huang et al., 2017a); GoogleNet (Szegedy et al., 2015); Inception (Szegedy et al., 2016); MobileNet (Howard et al., 2017); ResNet (He et al., 2016); SqueezeNet (Iandola et al., 2016); VGG (Simonyan and Zisserman, 2014).

Neste trabalho se decidiu fazer experimentos com os métodos de detecção de objetos *Faster* R-CNN, SSD e YOLOv3, variando algumas arquiteturas de CNN. Os modelos pré-treinados escolhidos foram:

- *Faster* R-CNN com Inception.
- *Faster* R-CNN com ResNet-50.
- *Faster* R-CNN com ResNet-101.
- SSD com MobileNet.
- SSD com ResNet-152.
- YOLOv3 com DarkNet-53.

A justificativa pela escolha desses modelos está na variação das características entre eles. Modelos de detecção de objetos baseados em região teoricamente possuem melhor valor de precisão média (AP), porém sua velocidade é bem inferior ao se comparar com um detector YOLOv3, por exemplo. O método SSD tenta fazer a interseção entre o valor da métrica AP e a velocidade de inferência. Além disso, arquiteturas de CNN MobileNets são desenvolvidas para sistemas embarcados, sendo bem mais enxutas do que uma arquitetura de CNN Inception, por exemplo. Com essa variação realizada nesse estudo, torna-se possível definir com mais precisão qual o melhor método de detecção de objetos e sua respectiva arquitetura de CNN conseguiu, com um razoável valor de AP e velocidade, reconhecer ovos da espécie *S. mansoni* em imagens microscópicas.

5.3 Experimentos e análise dos resultados

Com todos os modelos definidos foi possível dar início ao treinamento, com o objetivo de ensinar os modelos a reconhecerem ovos da espécie *S. mansoni* em imagens microscópicas. Cada um dos modelos possui seu próprio arquivo de configuração, em

que são definidos todos os hiperparâmetros para o treinamento. Encontrar os melhores hiperparâmetros de treinamento é uma tarefa extremamente complexa, por isso, muitos deles foram definidos empiricamente por meio de experimentos realizados e por valores já consolidados na literatura. Além disso, para fins comparativos, alguns valores dos hiperparâmetros foram ajustados igualmente para todos os modelos, de forma que fosse realizada uma comparação justa. A Figura 36 apresenta o diagrama do *pipeline* do processo para os experimentos realizados.

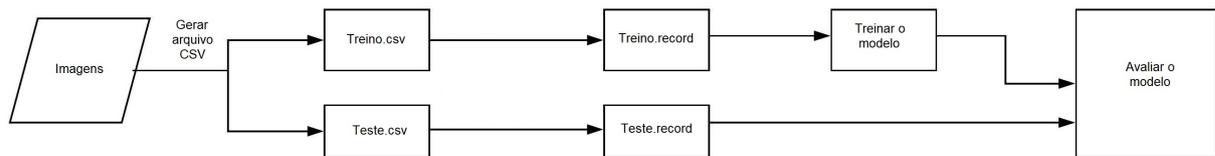


Figura 36 – Diagrama do *pipeline* de processo.

Empiricamente, variaram-se alguns valores de hiperparâmetros com o objetivo de encontrar a melhor configuração para os modelos. O número de classes de objetos a serem detectados foi definido como 1, uma vez que o foco do trabalho é encontrar ovos da espécie *S. mansoni*. O tamanho da entrada da imagem é um hiperparâmetro importante, pois o processo de redimensionamento gera perdas. Existem dois tipos: *fixed_shape_resize* e *keep_aspect_ratio_resizer*. O primeiro preenche a dimensão menor ao invés de distorcer a imagem, melhorando consideravelmente casos em que as imagens possuem uma variabilidade grande de resoluções. O segundo redimensiona a imagem de entrada mantendo a proporção do aspecto para satisfazer as restrições do tamanho mínimo e máximo definidos. Para os métodos baseados em dois estágios foi definido o *keep_aspect_ratio_resizer* com *min_dimension* e *max_dimension* ambos como 640. Para os detectores de um estágio, a altura e largura do *fixed_shape_resize* foram definidas como 640 e 640, respectivamente.

O hiperparâmetro *feature_extractor* define qual o extrator de recursos utilizado no modelo, tendo sido avaliado o Inception, ResNet-50, ResNet-152, MobileNet e DarkNet-53. Em todos os casos o aprendizado por transferência foi empregado, sendo que a rede foi totalmente retreinada. Os modelos foram anteriormente pré-treinados no conjunto de dados COCO (Lin et al., 2014). O hiperparâmetro *first_stage_features_stride* que descreve o passo de saída foi testado com dois valores: 16 e 8. Ao final, optou-se pelo valor 16. O *first_stage_nms_score_threshold* é o valor da pontuação para a Supressão Não Máxima (NMS) empregada para os detectores de dois estágios, os quais utilizam uma RPN. Esse

valor deve estar entre 0 e 1, pois é aplicado diretamente após uma transformação softmax. O valor aplicado para *Faster* R-CNN foi 0.1.

O limite de IoU para executar a NMS nas caixas previstas pela RPN foi definido como 0.7. Além desses hiperparâmetros exclusivos para o método *Faster* R-CNN, vale a pena lembrar que, como o *Faster* R-CNN tem duas redes, a primeira propõe regiões onde os objetos podem ser encontrados e a segunda tenta detectar objetos nessas redes, aumentar o número de propostas pela primeira rede aumenta a precisão, mas implica em mais trabalho computacional, pois a segunda rede tem que pesquisar em mais áreas potenciais. Sendo assim, o número de propostas a serem feitas pela RPN foi especificado como 300.

O uso de *Dropout* como técnica de regularização amplamente usada para evitar o risco de sobreajuste foi empregada em todos os modelos. O *batch_size* foi definido como 4 em todos os experimentos, mesmo tendo sido testado valores entre 1 e 16. Um *batch* maior que 4 estoura a memória disponível dado os recursos utilizados para o *Faster* R-CNN. Para o SSD e YOLOv3, foi possível ter memória suficiente para um *batch* de tamanho 16, porém, para fins comparativos, padronizou-se o valor de 4.

São três tipos de otimizadores disponíveis: RMSProp, momentum e adam. Para cada otimizador, é possível configurar a taxa de aprendizagem como: constante, decaimento exponencial, passo manual ou decaimento por cosseno. Por meio de experimentos realizados e consultas na atual literatura, os valores encontrados para o treinamento do modelo final foi definido como sendo o otimizador momentum com uma taxa de aprendizado por decaimento por cosseno para os seguintes atributos: *learning_rate_base* sendo 0.01, *warmup_learning_rate* em 0.001 e o *warmup_steps* sendo 5000. Além disso, para evitar que o gradiente exploda, foi definido por experimentação o valor de 10 para o hiperparâmetro do gradiente por norma.

O número de etapas foi definido como 250000 para todos os treinamentos. Por definição, o número de etapas não é uma época. O número de etapas de treinamento depende da configuração da taxa de aprendizagem e do tamanho do lote. O número de etapas geralmente é calculado como apresentado na Equação 5.1.

$$\frac{train_data_size}{batch_size} * epoch \quad (5.1)$$

Por fim, operações de aumento de dados online foram empregadas de maneira comum em todos os treinamentos, sendo elas: mudança de cor, saturação e contraste, corte na imagem, zoom, rotação e mudança de escala. Alguns outros ajustes que devem ser feitos no arquivo de configuração de cada um dos modelos se referem à caminhos do sistema, como a definição da pasta que contém as imagens para treinamento e teste. Os demais valores permaneceram como padrão pela documentação do TensorFlow, uma vez que tais valores tiveram bom desempenho em tarefas de *benchmark*.

Após treinados, cada um dos modelos foi avaliado nas métricas de avaliação para detecção de objetos no COCO. A Tabela 4 ilustra as métricas de avaliação consideradas, com destaque para a “*AP at IoU=.50 (PASCAL VOC metric)*”, principal métrica utilizada e apresentada como resultado em trabalhos que envolvem a tarefa de detecção de objetos. Para se entender como cada uma das métricas são obtidas, é necessário conhecer conceitos importantes como o de Intersecção Sobre União (IOU), verdadeiro positivo, falso positivo, falso negativo e verdadeiro negativo, além da métrica de precisão e revocação. Todos estes conceitos foram abordados no capítulo 2 e podem ser obtidos no trabalho de Padilla et al. (2020).

Tabela 4 – Métricas de avaliação para detecção de objetos em COCO.

<p>Precisão média (AP): AP com IoU=.50:.05:.95 (métrica do desafio principal) <i>AP com IoU=.50 (métrica do PASCAL VOC)</i> AP com IoU=.75 (métrica estrita)</p>
<p>AP por área: AP para objetos pequenos: área $<32^2$ AP para objetos médios: $32^2 < \text{área} < 96^2$ AP para objetos grandes: área $>96^2$</p>
<p>Revocação média (AR): AR recebeu 1 detecção por imagem AR recebeu 10 detecção por imagem AR recebeu 100 detecção por imagem</p>
<p>AR por área: AR para objetos pequenos: área $<32^2$ AR para objetos médios: $32^2 < \text{área} < 96^2$ AR para objetos grandes: área $>96^2$</p>

Dessa forma, torna-se possível avaliar a capacidade de generalização de cada um dos modelos, testando-os com imagens que não foram vistas durante o treinamento. Com isso, é possível identificar se realmente os modelos treinados conseguiram detectar os ovos,

além de obter um valor de AP que apresenta o nível de precisão dessas detecções. Além disso, foram realizadas várias inferências nas imagens de teste, de modo que foi possível observar as detecções sendo realizadas, bem como os ovos em que o modelo não conseguiu identificar.

Em seguida, com os resultados obtidos de detecção de cada um dos modelos no conjunto de imagens de teste, é possível definir com mais precisão qual o melhor método de detecção de objetos e sua respectiva arquitetura de CNN conseguiu, com um razoável valor de AP e velocidade, reconhecer ovos da espécie *S. mansoni* em imagens microscópicas.

5.3.1 Comparativo com imagens utilizadas nos trabalhos relacionados

Conforme anteriormente colocado, o foco do desenvolvimento da solução proposta é auxiliar o especialista no diagnóstico da esquistossomose, por meio da análise de imagens microscópicas. Para tal, torna-se necessária preparar as amostras com alguma técnica parasitológica. A maioria dos trabalhos relacionados não empregam a técnica de preparação Kato-Katz, a qual é a mais amplamente utilizada, por possuir um menor custo entre outras vantagens. Na solução proposta, empregou-se a técnica de Kato-Katz pensando principalmente no uso da solução em ambiente real, uma vez que o objetivo é de realmente auxiliar os especialistas na tomada de decisão, indo além de experimentos controlados.

Dessa forma, coletou-se 60 imagens de ovos da espécie *S. mansoni* de diferentes fontes (citadas nos trabalhos relacionados) em que as lâminas não foram preparadas utilizando a técnica Kato-Katz, com o objetivo de testar a solução aqui desenvolvida nessas imagens e verificar se seria possível obter resultados tão relevantes quanto os alcançados nos trabalhos relacionados.

Para realizar os experimentos, empregou-se a abordagem *k-fold* (Wong, 2015) para um $k = 5$. Dessa forma, para cada experimento, 48 imagens participaram do treinamento e 12 do teste, conforme apresentado na Figura 37. Para a configuração dos hiperparâmetros do modelo no treinamento foi utilizado os mesmos encontrados para o melhor modelo que utiliza imagens de Kato-Katz, descritos na seção anterior. Inclusive, as mesmas técnicas de aumento de dados *online* e *offline* foram empregadas, visto que o número de imagens pode ser considerado pequeno.

5-fold	Dataset				
Experiment 1	Test	Train	Train	Train	Train
Experiment 2	Train	Test	Train	Train	Train
Experiment 3	Train	Train	Test	Train	Train
Experiment 4	Train	Train	Train	Test	Train
Experiment 5	Train	Train	Train	Train	Test

Figura 37 – Estratégia *k-fold* utilizada para divisão do conjunto de dados.

5.4 Integração com modelos de aprendizado de máquina

Com o objetivo de auxiliar a solução baseada em aprendizado profundo, foi verificada a viabilidade quanto ao uso de modelos de aprendizado de máquina para ajudar no diagnóstico da esquistossomose. Para isso, os seguintes métodos foram considerados:

- *K - Nearest Neighbors.*
- *Naive Bayes.*
- *Random Forest.*
- *Support Vector Machine.*
- *XGBoost.*

Para a preparação da base de dados, os mesmos conjuntos de treino e teste divididos anteriormente foram mantidos. Em cada imagem do conjunto de treino anotada pelo especialista, foi realizada o corte na área da caixa delimitadora que envolve o ovo da espécie *S. mansoni*. Dessa forma, caso uma determinada imagem possuísse quatro ovos, por exemplo, quatro novas imagens seriam geradas pertencendo a classe “schisto”. Com isso, 1146 imagens com diferentes dimensões foram criadas, todas pertencentes a mesma classe, que contém um ovo *S. mansoni*. O mesmo procedimento foi realizado no conjunto de teste, resultando em um total de 56 imagens.

Para a classe “não schisto”, foi implementado um *script* que varria todas as imagens do treinamento e gerava aleatoriamente janelas na imagem com diferentes dimensões. O tamanho mínimo era de 150 *pixels* e o máximo de 250 *pixels*, para cada lado do retângulo.

Para não correr o risco de a janela sobrepor um ovo de *S. mansoni* na imagem, era verificado se cada uma das janelas não colidia com alguma caixa delimitadora anotada pelo especialista. Caso ocorresse essa situação, a janela aleatória gerada era descartada. Com o processo, 10260 imagens foram criadas. Para o conjunto de teste, o mesmo procedimento foi adotado, tendo sido geradas 740 imagens da classe “não schisto”. A Figura 38 ilustra o processo de geração das imagens para cada uma das classes abordadas, a partir de uma imagem original anotada pelo especialista.

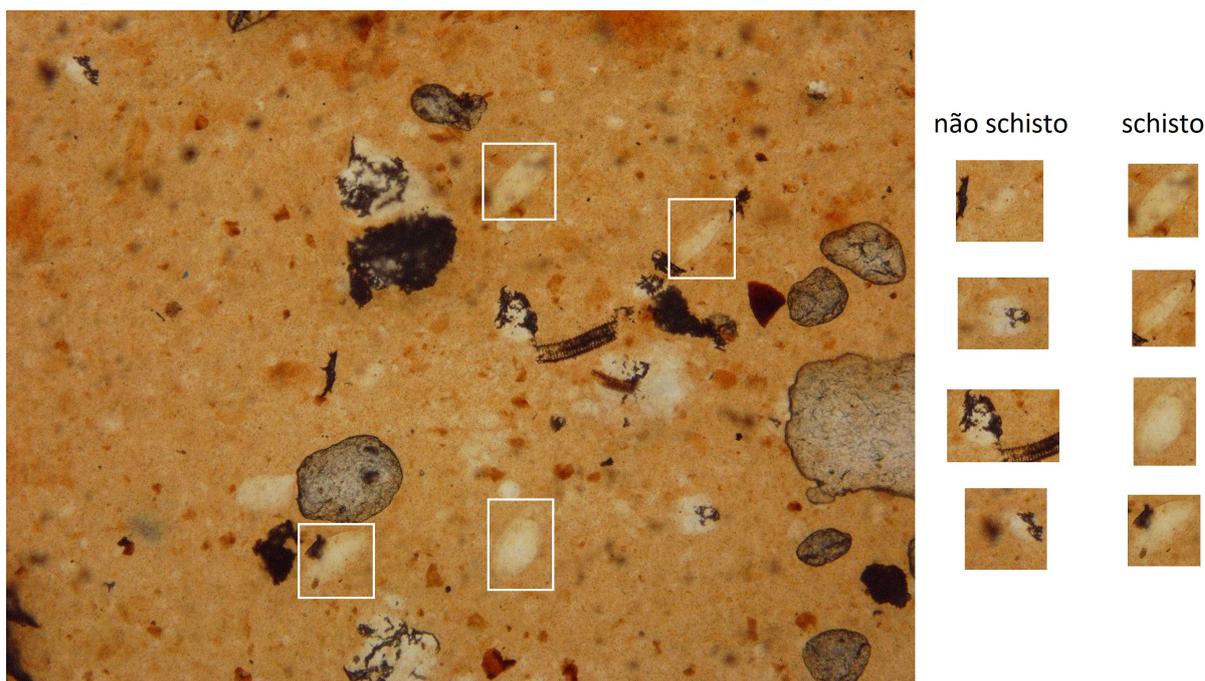


Figura 38 – Procedimento para criação do conjunto de imagens utilizado para o treinamento e teste dos métodos de aprendizado de máquina.

As caixas delimitadoras na cor branca sobre a imagem original mostradas na Figura 38 foram anotadas pelo especialista. A partir dos retângulos, foram gerados os cortes que fazem parte da classe “schisto”. Para a classe “não schisto”, caixas delimitadoras aleatórias foram geradas em diferentes posições da imagem, conforme anteriormente explicado.

Com o intuito de aumentar a quantidade de imagens para o treinamento dos métodos de aprendizado de máquina, para cada imagem do conjunto de treino pertencente a classe “schisto”, foi aplicada a operação de rotação em 90, 180 e 270 graus, além da inversão horizontal nas imagens originais e rotacionadas. Com isso, o número de imagens da classe “schisto” no conjunto de treino aumentou para 9168. A mesma abordagem foi empregada no conjunto de teste, resultando ao final do processo um total de 448 imagens.

Obtido o conjunto de treino e teste para cada uma das classes abordadas, o

descritor de recursos HOG foi aplicado. A técnica proposta no trabalho de Dalal and Triggs (2005) calcula as ocorrências da orientação dos gradientes de uma imagem. Para isso, o método usa a magnitude e o ângulo do gradiente para calcular os recursos, gerando diferentes histogramas locais para cada região da imagem. Dessa forma, o descritor se concentra na estrutura e no formato do objeto, conseguindo encontrar padrões específicos do objeto na imagem.

Como o retorno do uso da técnica HOG é um vetor das melhores características de cada imagem, o tamanho desse vetor é diretamente relacionado com o tamanho da imagem. Sabendo-se que a entrada dos classificadores clássicos de aprendizado de máquina é fixa, todas as imagens foram redimensionadas para um tamanho de 200×200 . Dessa forma, a saída da técnica HOG sempre resultava em um vetor com tamanho 1152, que representa os recursos mais relevantes de cada imagem.

O parâmetro usado para o filtro de reamostragem no processo de redimensionamento da imagem foi o *ANTIALIAS*, que consegue manter a qualidade da imagem com poucas perdas. Como parâmetros empregados para a técnica HOG, foi utilizado o valor 8 para o número de caixas por orientação, 16×16 como sendo o tamanho em *pixels* de uma célula e o valor 1×1 para o número de células em cada um dos blocos. A Figura 39 apresenta exemplos de imagens redimensionadas para as dimensões 200×200 e o resultado após a aplicação da técnica HOG.

A Figura 39 mostra exemplos de imagens do conjunto de treinamento da classe “schisto”, em que a imagem contém ovo da espécie *S. mansoni*. Já a Figura 40 ilustra exemplos de imagens do conjunto de treinamento da classe “não schisto”, em que a imagem não possui nenhum ovo da espécie *S. mansoni*, mas sim sujeiras, impurezas ou fungos.

Após a aplicação da técnica HOG em cada uma das imagens, quatro estruturas de conjunto de dados foram geradas: (1) classe “schisto” para treinamento com 9168 amostras e 1152 características; (2) classe “não schisto” para treinamento com 10260 amostras e 1152 características; (3) classe “schisto” para teste com 448 amostras e 1152 características; (4) classe “não schisto” para teste com 740 amostras e 1152 características.

Uma série de estudos (Provost, 2000; Batista et al., 2004; Chawla et al., 2004; He and Garcia, 2009; Ganganwar, 2012; He and Ma, 2013; Tyagi and Mittal, 2020; Vuttipittayamongkol et al., 2021) discute sobre os problemas enfrentados ao se trabalhar

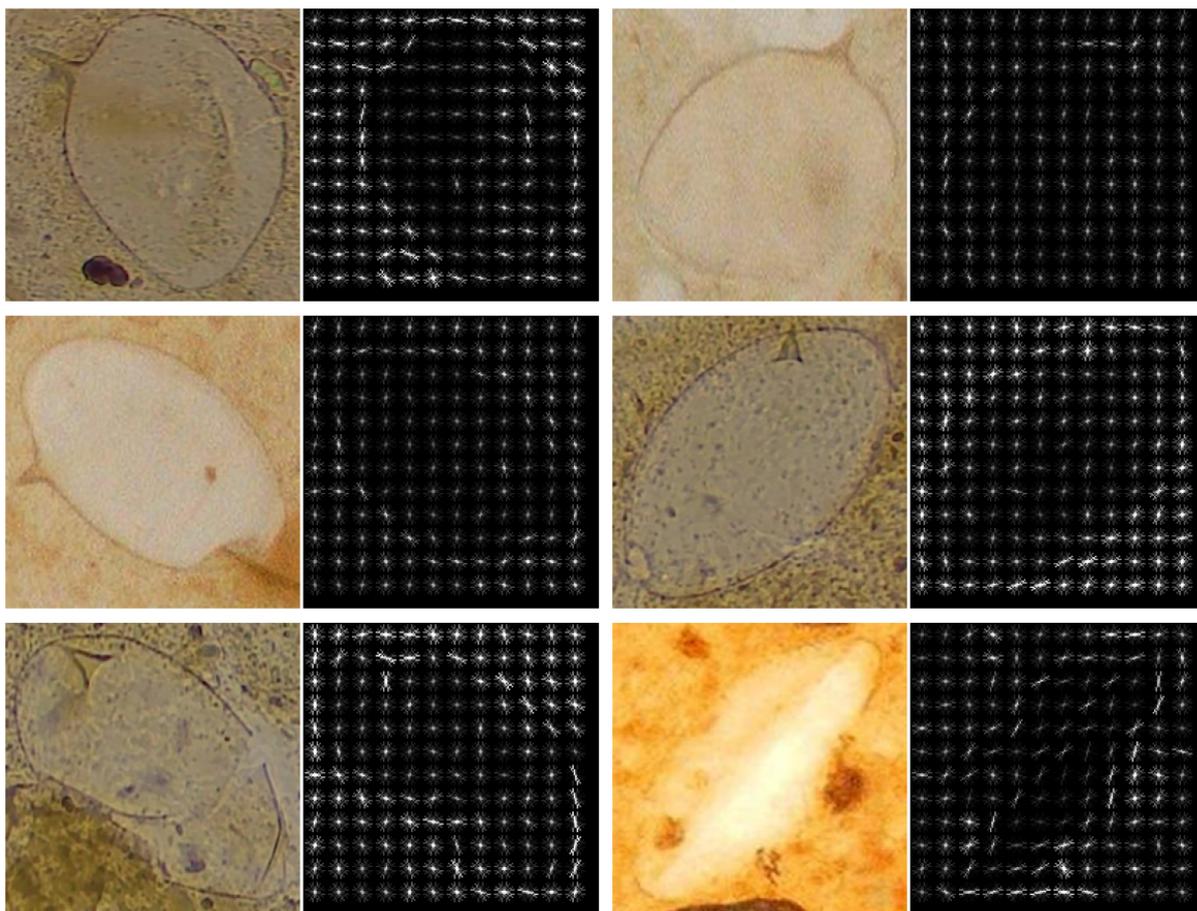


Figura 39 – Exemplos da aplicação da técnica HOG em imagens da classe “schisto”.

com classes de dados desbalanceadas para o treinamento de algoritmos de aprendizado de máquina. Por isso, abordagens foram empregadas nessa pesquisa, visando a diminuição do impacto causado por esses problemas.

Grande parte dos algoritmos de aprendizado de máquina funcionam melhor quando o número de amostras em cada classe possui aproximadamente a mesma quantidade. Isso ocorre porque a ideia geral desses algoritmos é maximizar a acurácia e reduzir os erros de classificação.

No entanto, se os dados forem desequilibrados, ou seja, a quantidade das classes não for próximas, então, em tais casos, métodos de aprendizado de máquina obtém uma acurácia muito alta apenas para a classe majoritária, não conseguindo classificar corretamente a classe minoritária.

Para o balanceamento dos dados, optou-se por fazer uma subamostragem nos dados da classe majoritária. A abordagem consiste em reduzir os dados eliminando amostras que pertencem à classe majoritária com o objetivo de igualar o número de amostras em

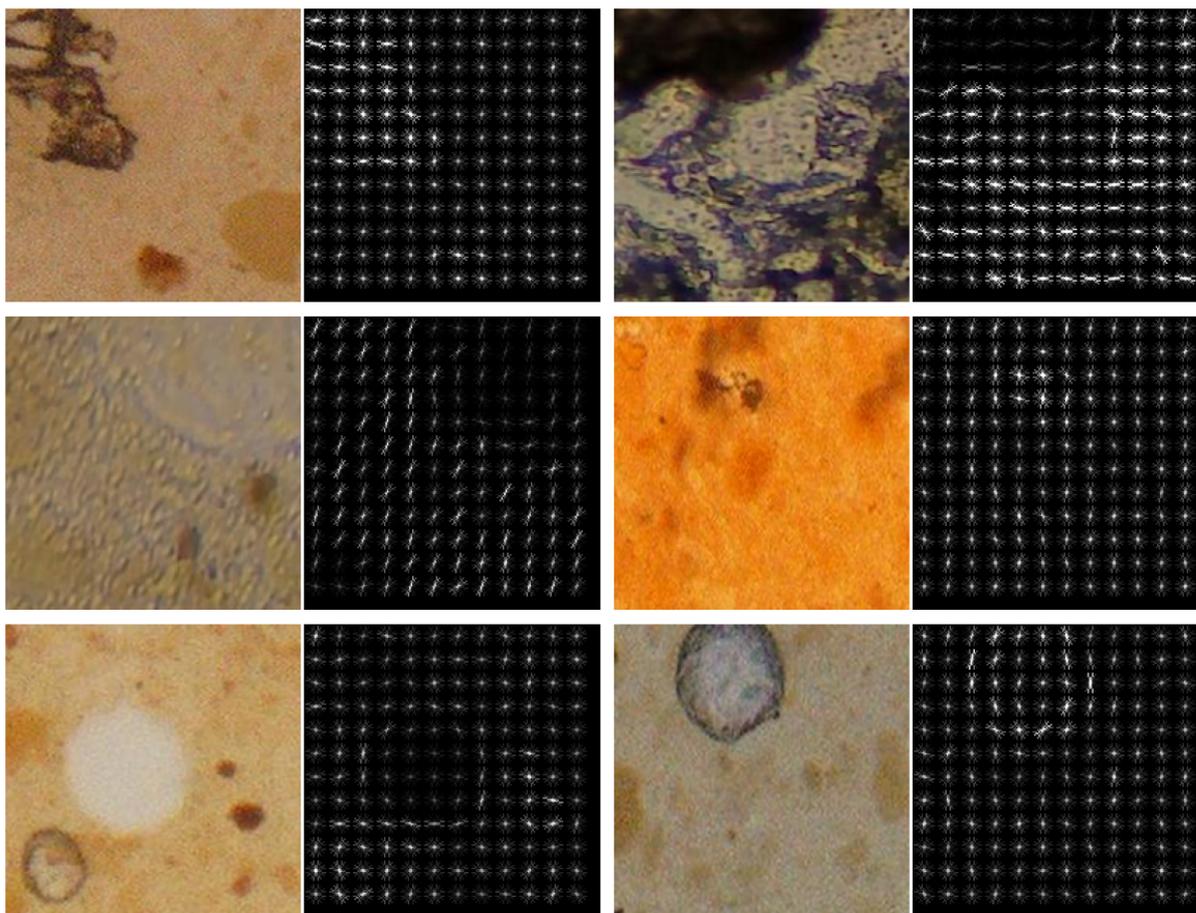


Figura 40 – Exemplos da aplicação da técnica HOG em imagens da classe “não schisto”.

cada classe. Uma técnica simples de subamostragem é a remoção aleatória de algumas amostras pertencentes à classe em maior quantidade. Contudo, a principal desvantagem desse método é que podem ser descartados dados potencialmente úteis que são importantes para o treinamento do algoritmo de aprendizado de máquina.

Por isso, abordagens de subamostragem não aleatórias foram empregadas nos dados. Tais técnicas utilizam de critérios baseados em heurísticas para a seleção de amostras redundantes ou que não agregariam durante o treinamento do modelo para que, após selecionadas, sejam excluídas. Os métodos utilizados foram o *Tomek Links* (Tomek, 1976) e a *Condensed Nearest Neighbors* (Hart, 1968). A combinação dessas duas técnicas foi proposta no trabalho de Kubat et al. (1997) e demonstrou ser promissora para os casos abordados.

O método propõe a aplicação da técnica *Tomek Links* e em seguida o uso da técnica *Condensed Nearest Neighbors*. As amostras selecionadas pela abordagem *Tomek Links* são consideradas redundantes e grande parte delas com muitos ruídos, enquanto no

método *Condensed Nearest Neighbors* é proposto a seleção de amostras que pertencem a classe majoritária e que estão distantes da curva de separação entre as classes. A Tabela 5 apresenta desde a quantidade inicial das imagens até o fim do processo de preparação dos dados para o treinamento e teste dos algoritmos de aprendizado de máquina abordados nessa pesquisa.

Tabela 5 – Preparação dos dados para os modelos de aprendizado de máquina.

	Treinamento		Teste	
	não schisto	schisto	não schisto	schisto
Original	10260	1146	740	56
Após aumento de dados	10260	9168	740	448
Após aplicação do HOG	10260	9168	740	448
Após aplicação do <i>Tomek Links</i>	9749	9168	571	448
Após aplicação do <i>Condensed Nearest Neighbors</i>	9168	9168	448	448

Como as imagens da classe “não schisto” foram geradas aleatoriamente e para fazer sentido o uso de algumas métricas de classificação que necessitam de os dados de teste estarem desbalanceados, a abordagem de subamostragem empregada nos dados do conjunto de treinamento também foi usada para os dados do conjunto de teste, removendo amostras não importantes da classe majoritária, que no caso, a “não schisto”.

Com o conjunto de dados balanceado e processado, deu-se início ao treinamento dos algoritmos de aprendizado escolhidos. Foi utilizada a scikit-learn, que é uma biblioteca de aprendizado de máquina de código aberto para a linguagem de programação Python (Pedregosa et al., 2011). Além disso, dado um conjunto de hiperparâmetros para cada modelo, foi investigado por meio da pesquisa em grade qual a melhor configuração para cada modelo, considerando os dados empregados nessa pesquisa. A Tabela 6 ilustra os hiperparâmetros testados para cada um dos modelos treinados.

Após o treinamento dos modelos e a seleção dos melhores hiperparâmetros para cada um deles, foi realizada a avaliação deles. As métricas empregadas foram: acurácia, precisão, revocação e F1-score. Com isso, foi possível compará-los e verificar se algum deles conseguiu de forma satisfatória classificar corretamente imagens que contém um ovo da espécie *S. mansoni*.

Tabela 6 – Parâmetros de treinamento dos modelos de aprendizado de máquina testados.

<i>K - Nearest Neighbors</i>	
n_neighbors	3, 5, 7, 9, 11, 13, 15
weights	uniform, distance
metric	euclidean, manhattan
<i>Naive Bayes</i>	
var_smoothing	1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7, 1e-8, 1e-9, 1e-10, 1e-11, 1e-12, 1e-13, 1e-14, 1e-15
<i>Random Forest</i>	
bootstrap	True, False
max_depth	10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None
max_features	auto, sqrt
min_samples_leaf	1, 2, 4
min_samples_split	2, 5, 10
n_estimators	200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000
<i>Support Vector Machine</i>	
C	0.1, 1, 10, 100, 1000
gamma	1, 0.1, 0.01, 0.001, 0.0001
kernel	linear, rbf, poly, sigmoid
<i>XGBoost</i>	
min_child_weight	1, 5, 10
gamma	0.5, 1, 1.5, 2, 5
subsample	0.6, 0.8, 1.0
colsample_bytree	0.6, 0.8, 1.0
n_estimators	400, 500, 600, 700
learning_rate	0.1, 0.01, 0.001
max_depth	3, 4, 5

Em seguida, foi feita uma análise criteriosa quanto a possibilidade dos modelos de aprendizado de máquina agregarem a solução proposta nesse estudo, auxiliando na redução do número de falsos positivos detectados pelo modelo de aprendizado profundo e aumentando a confiabilidade do sistema proposto, cujo objetivo é o diagnóstico da esquistossomose por meio de imagens microscópicas preparadas pela técnica Kato-Katz.

5.5 Comparativo com aplicativo comercial

Além dos experimentos realizados nesse estudo, foi realizado um comparativo entre a solução aqui desenvolvida com o ParasiteID (<https://parasite.id/>), um sistema *Web* que usa análise de imagens automatizada com aprendizado de máquina para facilitar o diagnóstico de infecção por vermes parasitários. A ferramenta ParasiteID foi desenvolvida

para ser acoplada a microscópios de telefones móveis e baratos, podendo ser operados por qualquer pessoa. De acordo com os desenvolvedores, a ferramenta de diagnóstico é altamente sensível e específica, reunindo o maior conjunto de dados de imagens de ovos de helmintos e técnicas de aprendizado profundo de última geração.

O funcionamento da ferramenta ParasiteID é bastante simples. O usuário realiza o *upload* de uma imagem microscópica de uma amostra de fezes e em seguida obtém uma previsão imediata sobre os ovos do parasito presentes na amostra. De acordo com os autores, o método foi avaliado e considerado mais de 99% preciso. Além disso, a ferramenta consegue identificar ovos das seguintes espécies: *Schistosoma haematobium*, *Schistosoma japonicum*, *Schistosoma mansoni*, *Ascaris lumbricoides*, *Fasciola hepatica*, *Ancylostoma duodenale*, *Necator americanus*, *Taenia sp.*, *Hymenolepis nana*, *Trichuris trichiura* e amostras sem nenhuma espécie.

5.6 Sistema para produção

Por fim, visando proporcionar a possibilidade do uso do sistema desenvolvido para os profissionais da área da saúde, foi implementado um aplicativo *Web* que permite que qualquer usuário faça o *upload* de uma imagem para ser analisada pela solução proposta, para que em seguida exiba os resultados para o profissional.

A aplicação *Web* foi desenvolvida utilizando o *framework* Flask, que é um módulo em Python que permite o desenvolvimento de aplicativos para a *Web* (Grinberg, 2018). Dessa forma, foi implementado um *endpoint* na *Application Programming Interface* (API) que recebe solicitações via o método POST suportado pelo protocolo HTTP.

Para realizar a requisição ao *endpoint* na API, foi proporcionada ao usuário uma interface gráfica para que ele consiga carregar uma imagem local e fazer o *upload* para a aplicação *Web*. Após a API receber a requisição, ela retorna as detecções realizadas pelo modelo de inteligência artificial e exibe para o usuário na interface gráfica os resultados obtidos.

Para que a aplicação seja utilizada é necessário o acesso por um navegador de internet. Caso o sistema esteja local, sempre que ele é iniciado, é criado um serviço para seu acesso, possuindo um endereço IP e uma porta. Quando iniciado, a ferramenta verifica se existe uma GPU disponível. Caso sim, o modelo é alocado na memória da GPU mas, caso

contrário, na memória RAM. Com isso, a API fica à disposição para receber requisições a qualquer momento, enquanto estiver *online*.

5.7 Resumo

Este capítulo descreveu os materiais e métodos utilizados nessa pesquisa. Primeiramente, criou-se o conjunto de imagens contendo ovos de parasitos da espécie *S. mansoni* e de algumas outras para que fosse testada a solução proposta. Posteriormente, foram definidos os materiais a serem utilizados e em seguida ocorreu a implementação dos métodos de detecção de objetos e suas arquiteturas de CNN escolhidas. Além disso, foram abordados diferentes métodos de aprendizado de máquina para a tarefa de classificação, com o objetivo de integrar e auxiliar os resultados de detecção no sistema. Por fim, foram realizados inúmeros experimentos buscando validar a solução proposta. Dessa forma, o próximo capítulo tem como objetivo apresentar os resultados obtidos nesse trabalho, expondo as consequências da realização de cada etapa descrita na metodologia.

Capítulo 6

Resultados e Discussão

No capítulo de Resultados e Discussão são apresentadas as informações coletadas e a análise dos resultados obtidos. São relatados os comportamentos dos métodos de inteligência artificial abordados nos experimentos feitos, quantificando os resultados. De uma forma genérica, a solução proposta nessa pesquisa pode ser observada na Figura 41.

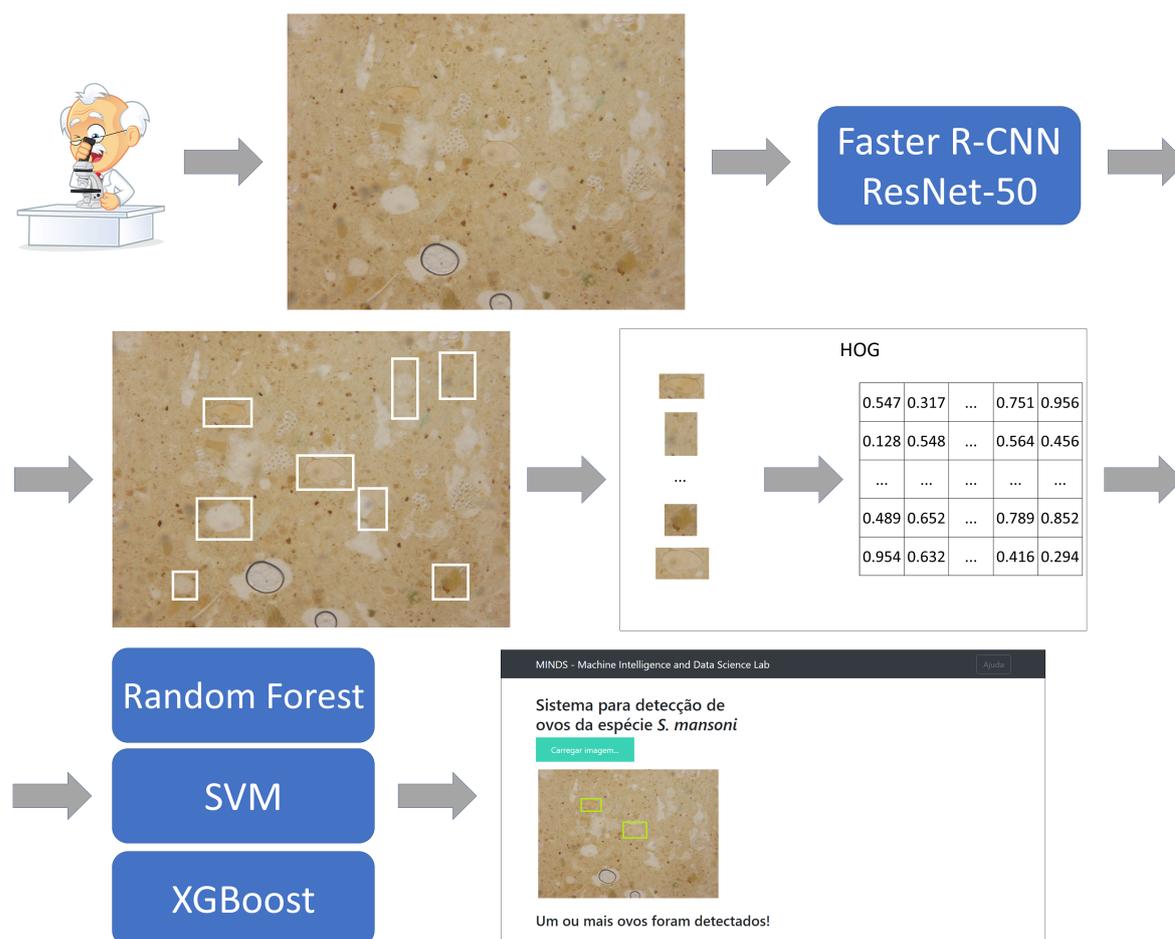


Figura 41 – Sistema proposto.

6.1 Comparativo entre as anotações dos especialistas

Como apontado no capítulo de Material e Métodos, inicialmente dois diferentes modelos de aprendizado profundo para detecção de objetos foram treinados com as diferentes anotações dos especialistas A e B. O método de detecção de objetos adotado foi o *Faster* R-CNN, com uma arquitetura de CNN ResNet-50 e outros dois treinamentos empregando a arquitetura Inception. A Tabela 7 expõe os resultados obtidos.

Tabela 7 – Resultados obtidos para o comparativo entre as anotações dos especialistas.

Métrica	IoU	área	maxDets	(1)	(2)	(3)	(4)
AP	0.50:0.95	all	100	0.215	0.634	0.136	0.569
	0.50			0.737	0.884	0.570	0.807
	0.75			0.019	0.763	0.010	0.722
	0.50:0.95	small		-1	-1	-1	-1
		medium		-1	-1	-1	-1
		large		0.216	0.625	0.139	0.587
AR	0.50:0.95	all	1	0.183	0.425	0.140	0.398
			10	0.333	0.708	0.274	0.693
			100	0.337	0.734	0.282	0.698
		small		-1	-1	-1	-1
		medium		-1	-1	-1	-1
		large	0.337	0.734	0.282	0.698	

onde,

1. *Faster* R-CNN com ResNet-50 treinado com as anotações do especialista A.
2. *Faster* R-CNN com ResNet-50 treinado com as anotações do especialista B.
3. *Faster* R-CNN com Inception treinado com as anotações do especialista A.
4. *Faster* R-CNN com Inception treinado com as anotações do especialista B.

Como pode ser percebido na Tabela 7, os resultados dos modelos treinados com as anotações realizadas pelo especialista B foram superiores em relação as marcações feitas pelo especialista A. Tanto os treinamentos realizados com o método de detecção de objetos *Faster* R-CNN empregando a arquitetura de CNN ResNet-50 e Inception utilizaram o mesmo conjunto de hiperparâmetros. Além disso, os modelos treinados foram avaliados nas mesmas imagens de teste, fazendo com que a comparação fosse justa.

Considerando um valor de AP @[IoU=0.50] (segunda linha da Tabela 7), o modelo *Faster R-CNN* com a arquitetura de CNN ResNet-50 treinado com as anotações do especialista B foi 14.7% superior em relação ao modelo treinado com as marcações do especialista A. Já para a arquitetura de CNN Inception, as anotações do especialista B usadas para o treinamento fez com que o modelo obtivesse um resultado de 23.7% a mais do que o alcançado pelo modelo treinado com as marcações do especialista A. Além disso, também foi considerado o desvio padrão das áreas calculadas para cada uma das caixas delimitadores geradas por cada um dos especialistas, além da dispersão dos dados das anotações, conforme explicado no capítulo de Material e Métodos.

6.2 Métodos de detecção de objetos

Dessa forma, após a análise dos resultados dos experimentos iniciais, foi definido o conjunto de imagens anotadas pelo especialista B para o treinamento de outros modelos de aprendizado profundo para detecção de objetos. A Tabela 8 apresenta os valores de cada experimento realizado para cada um dos métodos de detecção de objetos abordados no estudo.

Tabela 8 – Resultados obtidos para os experimentos com diferentes métodos de detecção de objetos.

Métrica	IoU	área	maxDets	(1)	(2)	(3)	(4)	(5)
AP	0.50:0.95	all	100	0.569	0.634	0.648	0.620	0.363
	0.50			0.807	0.884	0.838	0.865	0.624
	0.75			0.722	0.763	0.785	0.755	0.377
	0.50:0.95	small		-1	-1	-1	-1	-1
		medium		-1	-1	-1	-1	-1
		large		0.587	0.625	0.662	0.620	0.363
AR	0.50:0.95	all	1	0.398	0.425	0.427	0.391	0.272
			10	0.693	0.708	0.713	0.694	0.497
		100	all	0.698	0.734	0.728	0.715	0.574
				small	-1	-1	-1	-1
			medium	-1	-1	-1	-1	-1
			large	0.698	0.734	0.728	0.715	0.574

onde,

1. *Faster R-CNN* com Inception.
2. *Faster R-CNN* com ResNet-50.

3. *Faster* R-CNN com ResNet-101.
4. SSD com MobileNet.
5. SSD com ResNet-152.

Para o método YOLO com DarkNet-53, foi medido apenas o valor de **AP @[IoU = 0.50 | area = all | maxDets = 100]**, que resultou em 0.661. Como era esperado, o método YOLO que é baseado em somente um estágio, não conseguiu obter um valor tão bom quanto os baseados em dois estágios, que é o caso do método *Faster* R-CNN. Porém, a mesma observação não é válida para o método de detecção de objetos SSD com uma arquitetura de CNN MobileNet, que obteve um valor de AP @[IoU=0.50] de 0.865, superando alguns modelos treinados usando o método de detecção de objetos *Faster* R-CNN, como os das arquiteturas de CNN Inception e ResNet-101.

Em resumo, a precisão mede a taxa de falsos positivos ou a proporção de detecções de objetos verdadeiros para o número total de objetos que o classificador previu. Revocação mede a taxa de falsos negativos ou a proporção de detecções de objetos verdadeiros para o número total de objetos no conjunto de dados. Ao analisar a AP @[IoU=0.50] (segunda linha da Tabela 8), obteve-se um valor interessante (≥ 0.8) para a detecção dos ovos da espécie de *S. mansoni* para alguns modelos treinados.

Ao se observar a Tabela 8 fica evidente que os resultados de teste empregando o método de detecção de objetos *Faster* R-CNN com ResNet-50 foram superiores aos demais comparados, o que era de se esperar. O método *Faster* R-CNN possui uma abordagem com dois estágios e tem como principal foco a precisão nas detecções. Além disso, a arquitetura de CNN ResNet-50 possui uma estrutura muito bem consolidada e é utilizada em várias tarefas na área da visão computacional para a classificação de imagens. Dessa forma, com base na métrica AP, foi definido que o método de detecção de objetos que será parte da solução é o *Faster* R-CNN, com uma arquitetura de CNN ResNet-50.

Ao se comparar as arquiteturas de CNN ResNet e Inception, tem-se que ambas são redes poderosas e com muitas camadas de convolução, o que pode ocasionar o problema do desaparecimento do gradiente. Dessa forma, foi observado que a arquitetura ResNet tende a reduzir esse problema durante o treinamento, uma vez que ela possui conexões

diretas que saltam algumas camadas. Sendo assim, graças a essas conexões com atalho, percebeu-se uma maior fluidez do gradiente durante os experimentos realizados.

Encontrar um otimizador e uma taxa de aprendizado adequado foi um desafio. Quando foi utilizada uma taxa de aprendizado muito pequena a convergência foi extremamente lenta, enquanto ao se utilizar uma grande fez com que a função de perda flutuasse em torno do mínimo local, não convergindo. Além disso, percebeu-se que o otimizador *momentum* acumula o gradiente das etapas anteriores para determinar a qual direção ele deve ir, o que é uma vantagem.

Por fim, percebeu-se que foi possível evitar o *overfitting* e obter um melhor desempenho do modelo por meio de abordagens avançadas para controlar a taxa de aprendizagem ao longo do tempo. Em especial, empregando o cosine decay, foi observado que durante o treinamento foi encontrado um bom mínimo local para a função de perda.

6.3 Seleção do limiar de detecção para o modelo de detecção de objetos

Com o objetivo de encontrar o melhor limiar de detecção para o modelo, a Tabela 9 apresenta os resultados obtidos dos experimentos realizados com as imagens de teste para diferentes limiares de detecção, enquanto a Figura 42 ilustra as mesmas informações, dessa vez, em forma de gráfico, facilitando a visualização e entendimento. O eixo Y mostra a quantidade de falsos negativos enquanto o eixo X representa os falsos positivos.

Tabela 9 – Variação dos limiares de detecção para o modelo *Faster* R-CNN com uma arquitetura de CNN ResNet-50.

Limiar	Ovo detectado	Falso negativo	Falso positivo	Sensibilidade
0.1	48	8	35	85.71%
0.2	45	11	11	80.35%
0.3	44	12	8	78.57%
0.4	44	12	4	78.57%
0.5	44	12	3	78.57%
0.6	43	13	1	76.78%
0.7	38	18	1	67.85%
0.8	32	24	0	57.14%
0.9	22	34	0	39.28%

O valor de especificidade e o gráfico da curva ROC (*Receiver Operating Characteristic*) não foram apresentados por não fazerem sentidos em tarefas de detecção de objetos. Para se calcular a taxa de falsos positivos é necessário que se conheça os verdadeiros negativos, o que não se aplica, pois representaria um erro de detecção corrigido. Nesse tipo de tarefa existem muitas caixas delimitadoras possíveis que não devem ser detectadas em uma imagem. Assim, os verdadeiros negativos seriam todas as caixas delimitadoras possíveis que não foram detectadas corretamente, não fazendo sentido o cálculo dessa métrica.

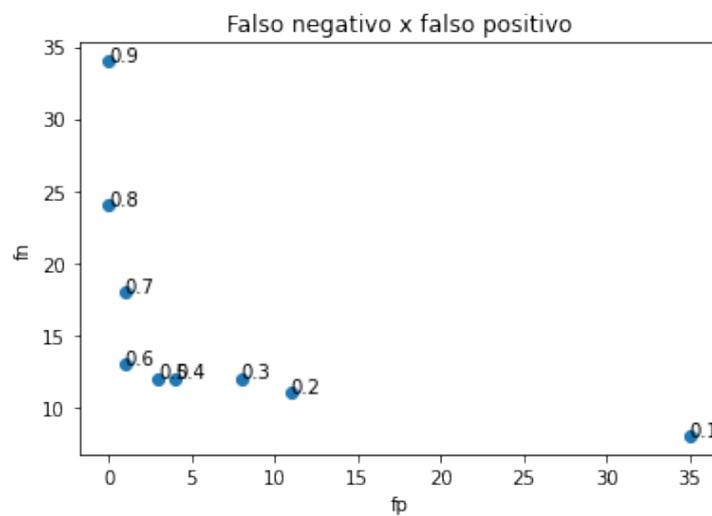


Figura 42 – Gráfico da variação dos limiares de detecção para o modelo *Faster R-CNN* com uma arquitetura de CNN ResNet-50.

Como é possível perceber nos dados mostrados na Tabela 9 e na Figura 42, os valores mostram um problema clássico de *trade off*, uma vez que diminuir o limiar do modelo impacta em uma maior quantidade de ovos da espécie *S. mansoni* detectados, minimizando o número de falsos negativos, porém, implicando em uma maximização dos falsos positivos. Em contrapartida, quando o limiar de detecção cresce, consequentemente reduzindo a sensibilidade do modelo, o número de falsos positivos cai, porém, a quantidade de falsos negativos aumenta.

A Figura 43 apresenta o gráfico exposto na Figura 42 mas, dessa vez, com os valores normalizados.

O ponto em vermelho representa o ponto de utopia, também conhecido como solução ideal, dado que o ponto é formado pelas coordenadas mínimas que minimizam os dois objetivos ilustrados no gráfico. O ponto em azul ilustra a melhor solução dentre as

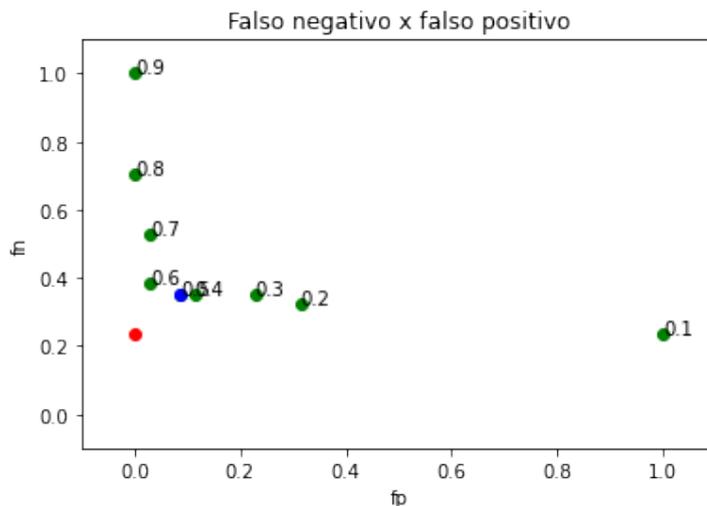


Figura 43 – Gráfico normalizado da variação dos limiares de detecção para o modelo *Faster R-CNN* com uma arquitetura de CNN ResNet-50.

possíveis, que é um limiar de detecção de 50%, considerando que tanto o falso negativo quanto o falso positivo possuem o mesmo peso. Para a conclusão da melhor solução, foi calculada a distância euclidiana entre cada ponto representado por cada um dos limiares adotados em relação ao ponto de utopia, com a menor distância sendo para o valor de limiar em 0.5.

6.4 Métodos de aprendizado de máquina

Com relação aos treinamentos dos classificadores *K - Nearest Neighbors*, *Naive Bayes*, *Random Forest*, *Support Vector Machine* e *XGBoost*, o conjunto dos melhores hiperparâmetros para o KNN foram: *metric*: euclidean, *n_neighbors*: 3, *weights*: uniform. A Figura 44 ilustra os resultados da matriz de confusão obtidos com a inferência do modelo KNN nas imagens do conjunto de teste. A acurácia do modelo foi de 67.63%.

Como é possível perceber na Figura 44, o modelo de aprendizado de máquina KNN conseguiu acertar 93% da classe “não schisto”, ou seja, previu corretamente o rótulo “0” quando ele realmente era “0”. Em contrapartida, pode-se reparar com os resultados apresentados que, em 58% das vezes, o modelo KNN rotula como “não schisto”, o que deveria ser rotulado como sendo um ovo, representado como a classe “schisto” (valor “1”). Sendo assim, o modelo KNN não se torna uma alternativa viável para esse tipo de aplicação utilizando as abordagens apresentadas, uma vez que o número de falsos negativos é considerável.

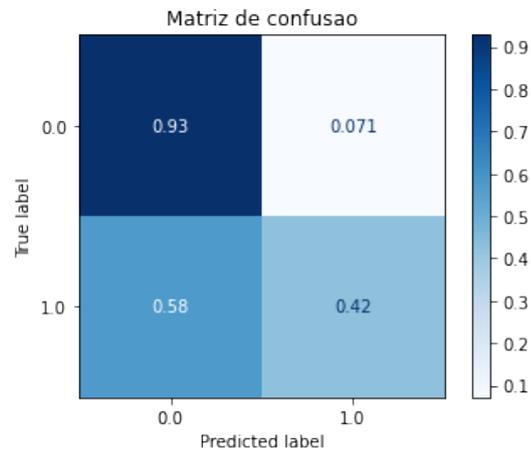
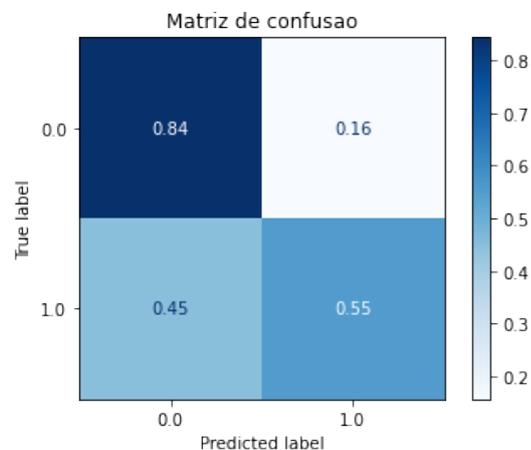


Figura 44 – Matriz de confusão do modelo KNN.

Para o modelo de aprendizado de máquina *Naive Bayes*, pela sua simplicidade, o único hiperparâmetro configurável e que teve o melhor desempenho entre as configurações propostas, foi o *var_smoothing* com o valor de 0.01. O modelo conseguiu uma acurácia de 69.75% no conjunto de teste e a Figura 45 apresenta os resultados da matriz de confusão obtidos.

Figura 45 – Matriz de confusão do modelo *Naive Bayes*.

Assim como o modelo de aprendizado de máquina KNN, o modelo *Naive Bayes* obteve uma taxa de falso negativo considerável. Isso quer dizer que o modelo, em grande parte das vezes, tende a classificar a imagem como sendo da classe “não schisto”, mesmo ela pertencendo a classe “schisto”. Para o problema proposto no estudo que é a detecção correta de ovos da espécie de *S. mansoni* em imagens microscópicas de lâminas preparadas pela técnica Kato-Katz, a avaliação dessa taxa de falso negativo é extremamente relevante, considerando que o sistema proposto tem que conseguir identificar corretamente tais ovos.

Em relação ao treinamento realizado com o modelo *Random Forest*, o melhor conjunto de hiperparâmetros obtido foi o seguinte: *bootstrap*: False, *max_depth*: 80, *max_features*: auto, *min_samples_leaf*: 1, *min_samples_split*: 5, *n_estimators*: 1200. A Figura 46 mostra os resultados da matriz de confusão obtidos com a inferência do modelo *Random Forest* nas imagens do conjunto de teste. A acurácia do modelo foi de 80.35%.

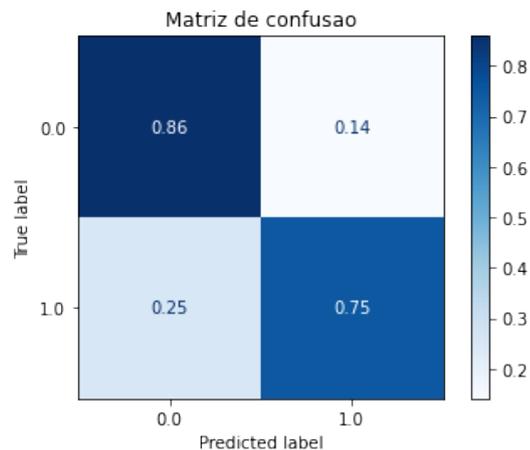


Figura 46 – Matriz de confusão do modelo *Random Forest*.

Ao ser comparado com os modelos KNN e *Naive Bayes* anteriormente apresentados, é possível perceber que os resultados alcançados pelo modelo de aprendizado de máquina *Random Forest* foram consideravelmente superiores. De todas as amostras rotuladas como pertencentes a classe “schisto” (rótulo 1), o modelo previu corretamente em 75% dos casos. Para a classe “não schisto” (rótulo 0), o *Random Forest* conseguiu manter uma precisão aproximada com a obtida pelos outros dois modelos, tendo obtido uma taxa de acerto de 86%.

Para o modelo de aprendizado de máquina SVM, a melhor configuração de hiperparâmetros foi: *C*: 10, *gamma*: 0.1, *kernel*: rbf. O modelo conseguiu uma acurácia de 82.81% no conjunto de teste e a Figura 47 apresenta os resultados da matriz de confusão obtidos.

Superando todos os modelos de aprendizado de máquina anteriormente apresentados, o método SVM conseguiu classificar corretamente em 89% das vezes que a imagem pertencia a classe “não schisto” e 77% das amostras pertencentes a classe “schisto”. A combinação entre a técnica de extração de recursos HOG com o modelo SVM é empregada com sucesso em vários trabalhos encontrados na literatura para a classificação de imagens (Cao et al., 2011; Yao et al., 2014; Lee et al., 2015; Yadav et al., 2015; Kong et al., 2019;

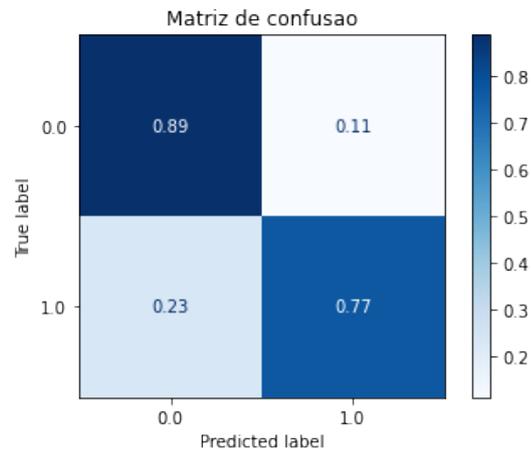
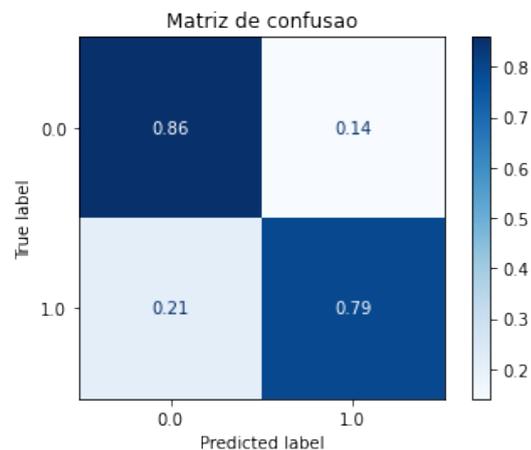


Figura 47 – Matriz de confusão do modelo SVM.

Xing et al., 2019), o que ajuda a explicar os resultados interessantes obtidos também nesse estudo.

Em relação ao treinamento realizado com o modelo *XGBoost*, o melhor conjunto de hiperparâmetros obtido foi o seguinte: *min_child_weight*: 1, *gamma*: 0.5, *subsample*: 1, *colsample_bytree*: 1, *n_estimators*: 400, *learning_rate*: 0.1, *max_depth*: 5. A Figura 48 mostra os resultados da matriz de confusão obtidos com a inferência do modelo *XGBoost* nas imagens do conjunto de teste. A acurácia do modelo foi de 82.47%.

Figura 48 – Matriz de confusão do modelo *XGBoost*.

Ao comparar os resultados obtidos do modelo *XGBoost* com os obtidos pelos demais modelos de aprendizado de máquina apresentados anteriormente, o *XGBoost* conseguiu os melhores resultados, conforme apresentados na matriz de confusão da Figura 48. O método *XGBoost* foi o que melhor conseguiu minimizar a taxa de falsos positivos e falsos negativos, considerando os dois objetivos simultaneamente. O modelo classificou

corretamente as imagens do conjunto de teste 79% das amostras que possuíam um ovo da espécie *S. mansoni*. Para as imagens que não possuíam um ovo da espécie *S. mansoni*, podendo ser um fungo, uma bolha, qualquer tipo de sujeira, impureza, ou até mesmo um ovo de outra espécie, o modelo conseguiu acertar em 86% dos casos.

A Tabela 10 apresenta os resultados obtidos por cada um dos modelos de aprendizado de máquina adotados no estudo. Todos foram submetidos ao mesmo conjunto de imagens de teste, após terem tido suas melhores características selecionadas por meio da técnica de extração de recursos HOG.

Tabela 10 – Resultados obtidos pelos modelos de aprendizado de máquina treinados para a tarefa de classificação.

Modelo	Acurácia	Precisão	Revocação	F1-score
KNN	67.63%	92.91%	42.00%	74.07%
<i>Naive Bayes</i>	69.75%	84.00%	55.00%	73.36%
<i>Random Forest</i>	80.35%	86.00%	75.00%	81.52%
SVM	82.81%	89.00%	77.00%	83.96%
<i>XGBoost</i>	82.47%	86.00%	79.00%	83.09%

Pelo fato de as classes do conjunto de teste estarem balanceadas, isto é, são 448 imagens pertencentes a classe “não schisto” e 448 imagens que pertencem a classe “schisto”, a avaliação da métrica acurácia torna-se viável. Como pode ser observado na Tabela 10, o modelo que obteve um melhor valor de acurácia foi o SVM, tendo obtido 82.81% das classificações corretas, independentemente da classe. Os classificadores *XGBoost* e *Random Forest* ocuparam a segunda e terceira posição no ranking, considerando a métrica acurácia. O modelo *XGBoost* alcançou uma acurácia de 82.47% enquanto o *Random Forest* 80.35%. Os métodos *Naive Bayes* e KNN obtiveram resultados consideravelmente inferiores que os seus concorrentes, não fazendo sentido continuar explorando esses dois últimos para o problema do estudo.

Apesar da avaliação empregando a métrica acurácia ser viável para comparar os modelos de classificação obtidos, ela sozinha não é o bastante para um comparativo criterioso. A métrica de revocação apresentada na Tabela 10 indica a porcentagem das vezes em que o modelo conseguiu classificar corretamente como “schisto”, as imagens que realmente continham um ovo da espécie *S. mansoni*. Dessa forma, ao ser avaliada essa

métrica, é possível perceber que o método de aprendizado de máquina que obteve o melhor desempenho foi o *XGBoost*, seguido pelos modelos SVM e *Random Forest*, respectivamente.

6.5 Integração entre o modelo de detecção de objetos e os modelos de aprendizado de máquina

Como o objetivo principal dos modelos de aprendizado de máquina é auxiliar na redução dos falsos positivos detectados pelo método *Faster R-CNN* com uma arquitetura de CNN ResNet-50, é importante que tais modelos consigam detectar corretamente quando um ovo pertencer a classe “schisto”, para que o método de aprendizado de máquina não elimine o ovo detectado incorretamente, e assim gerando o erro do falso negativo.

Devido os resultados apresentados de cada um dos classificadores, decidiu-se levar em consideração os resultados obtidos pelos modelos *Random Forest*, SVM e *XGBoost* para a decisão final de se a detecção do *Faster R-CNN* com ResNet-50 é um ovo de *S. mansoni* ou se é um falso positivo. Portanto, a integração de ambos os métodos ocorreu da seguinte forma:

1. O método de detecção de objetos *Faster R-CNN* com uma ResNet-50 faz as detecções.
2. Para cada caixa delimitadora encontrada, é avaliada a confiança que o método de aprendizado profundo tem em relação a aquela detecção.
3. Caso a confiança para a detecção seja superior a um determinado “limite de confiança”, a solução proposta mantém o ovo detectado pelo *Faster R-CNN*.
4. Para cada caixa delimitadora encontrada e que a confiança para a detecção seja inferior a um determinado “limite de confiança”, é feita a extração de recursos por meio da técnica HOG.
5. Cada um dos modelos *Random Forest*, SVM e *XGBoost* faz a inferência dos recursos extraídos.
6. Por critério de votação, a classe (“não schisto” ou “schisto”) que conseguir mais votos é considerada.

7. Caso a classe “não schisto” seja a mais votada, a caixa delimitadora em torno do objeto encontrado pelo *Faster R-CNN* é descartada, fazendo com que essa detecção seja considerada um falso positivo.
8. Caso a classe “schisto” seja a mais votada, a caixa delimitadora em torno do objeto encontrada pelo *Faster R-CNN* é mantida, fazendo com que essa detecção seja considerada um verdadeiro positivo.

A Tabela 11 apresenta os 10 melhores (em ordem) resultados obtidos por meio da combinação de diferentes limiares de detecção para o método de aprendizado profundo *Faster R-CNN* com uma ResNet-50 integrado aos classificadores de aprendizado de máquina.

Tabela 11 – Variação dos limiares de detecção para o modelo *Faster R-CNN* integrado aos classificadores de aprendizado de máquina.

Limiar Faster R-CNN	Limite de confiança	Ovo detectado	Falso negativo	Falso positivo	Sensibilidade
0.1	0.1	48	8	35	85.71%
0.1	0.5	46	10	11	82.14%
0.1	0.4	46	10	11	82.14%
0.1	0.3	46	10	14	82.14%
0.1	0.2	46	10	16	82.14%
0.2	0.5	45	11	6	80.35%
0.2	0.4	45	11	6	80.35%
0.2	0.3	45	11	9	80.35%
0.2	0.2	45	11	11	80.35%
0.2	0.1	45	11	11	80.35%

A primeira coluna da Tabela 11 mostra a variação do limiar de detecção do método de detecção de objetos *Faster R-CNN*, ou seja, é o valor que o método considera ou descarta a detecção, dado todas as caixas delimitadores geradas na imagem no momento da inferência. Já a segunda coluna da Tabela 11 expõe o limite de confiança para que o sistema proposto considere ou não utilizar os métodos de aprendizado de máquina. Por exemplo: supondo que o *Faster R-CNN* tenha encontrado somente um ovo cuja confiança é de 80%, caso o “limite de confiança” seja de 0.5, o sistema de diagnóstico irá considerar como um ovo da espécie *S. mansoni*. Para essa mesma detecção, caso o “limite de confiança”

seja de 0.9, os métodos de aprendizado de máquina *Random Forest*, SVM e *XGBoost* irão tomar a decisão de manter o objeto encontrado ou de excluir a caixa delimitadora gerada.

Como mostrado e discutido anteriormente na Tabela 9, o método de detecção de objetos *Faster R-CNN* com uma ResNet-50 sem o auxílio dos modelos de aprendizado de máquina conseguiu, dado um limiar de 10%, detectar corretamente 85.71% dos ovos da espécie *S. mansoni*. Entretanto, gerou uma quantidade (35 “não schisto”) muito alta de falsos positivos, o que é um problema, uma vez que pode diminuir a credibilidade da solução junto aos especialistas. Além disso, detectar incorretamente muitas áreas que não contém ovos da espécie *S. mansoni* pode confundir o especialista, fazendo com que o diagnóstico final esteja equivocado.

Ao observar a Tabela 11, percebe-se que a melhor combinação (considerando somente a sensibilidade) foi com um limite de confiança baixo (0.1). Isso quer dizer que para essa configuração os modelos de aprendizado de máquina *Random Forest*, SVM e *XGBoost* nem foram considerados, uma vez que sempre se manteve somente o resultado da detecção realizada pelo *Faster R-CNN* com uma ResNet-50. Por esse motivo os valores de falsos positivos e falsos negativos foram os mesmos (Tabela 9 e 11), uma vez que os modelos de aprendizado de máquina não foram utilizados.

Como apresentado na Tabela 10 e nas matrizes de confusão (Figuras 44, 45, 46, 47 e 48) obtidas pelos métodos de aprendizado de máquina, foi visto que os modelos conseguem ser melhores em classificar corretamente a classe “não schisto”, o que gera falsos negativos. Em contrapartida, ao se tratar do método de detecção de objetos *Faster R-CNN* com um limiar de detecção baixo, tem-se um valor considerável de falsos positivos, contudo, os ovos que aparecem na imagem são detectados corretamente. Dessa forma, a integração dessas duas técnicas pode contribuir positivamente com a solução proposta, uma vez que são complementares.

Como uma alternativa apresentada na Tabela 11, tem-se a utilização do limiar de detecção do *Faster R-CNN* em 0.1 e o limite de confiança para o auxílio dos modelos de aprendizado de máquina em 0.5. Um ponto negativo para essa combinação foi de que, em dois casos, o método de detecção de objetos *Faster R-CNN* encontrou corretamente os ovos, entretanto, os modelos de aprendizado de máquina consideraram não ser um ovo da espécie da espécie *S. mansoni*, excluindo a caixa delimitadora e não levando em conta a

detecção. A Figura 49 ilustra justamente esses dois casos.

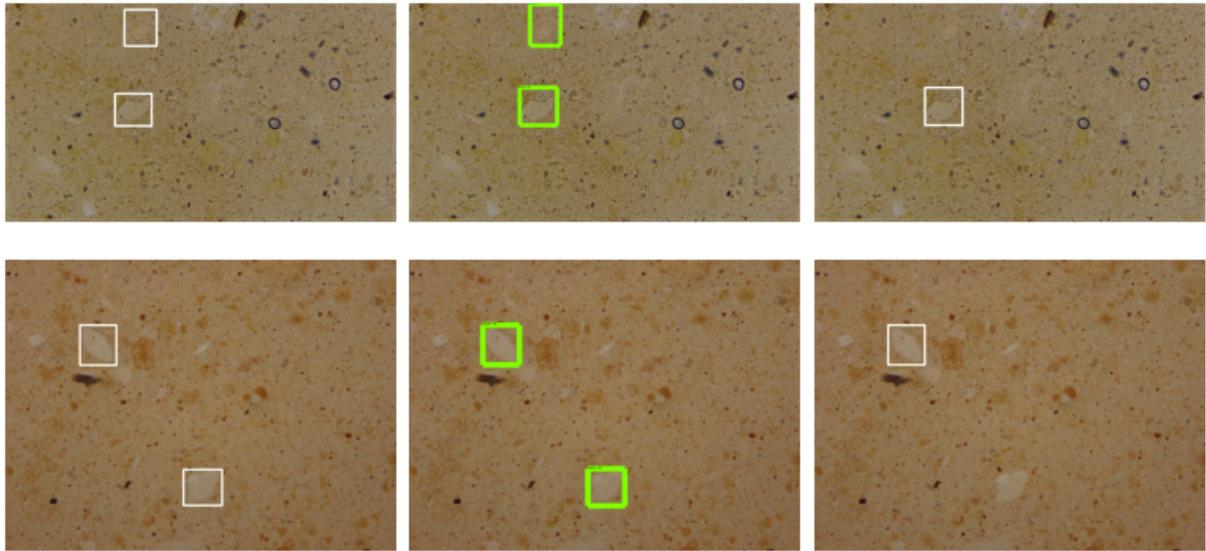


Figura 49 – Exemplo de casos em que a integração entre o detector de objetos de aprendizado profundo e os modelos de aprendizado de máquina atrapalhou no diagnóstico final da esquistossomose.

Na primeira coluna da Figura 49 é mostrada a imagem do teste e sua marcação original. Na segunda coluna é apresentada as marcações encontradas somente pelo método *Faster R-CNN* com uma ResNet-50 sem o auxílio dos métodos de aprendizado de máquina. Já a última coluna expõe o resultado da solução proposta. As imagens possuem dimensões diferentes pois são dois casos em que foi usado modelos de microscópios distintos. O lado positivo para ambos os casos discutidos é que, apesar da exclusão de uma marcação gerada pelo *Faster R-CNN*, ainda assim a solução final daria como resultado que foi encontrado um ovo da espécie *S. mansoni*.

A Figura 50 mostra os dois casos analisados anteriormente, considerando a anotação realizada pelo especialista A, pelo especialista B e pelo especialista C.

Interessante observar na primeira coluna da Figura 50 a não marcação justamente dos ovos que foram considerados como pertencentes a classe “não schisto” dos modelos de aprendizado de máquina. Nesses casos, o especialista também considerou que a região não se tratava de um ovo da espécie *S. mansoni*, mas sim qualquer outra coisa. A anotação dessa região foi considerada no conjunto de teste pelo fato de que, um terceiro especialista, assim como o segundo, considerou os objetos como um ovo de *S. mansoni*. Dessa forma, faz sentido concluir que os casos em que a solução proposta falhou, foram justamente os mesmos ovos que um especialista treinado também não iria considerar em um ambiente

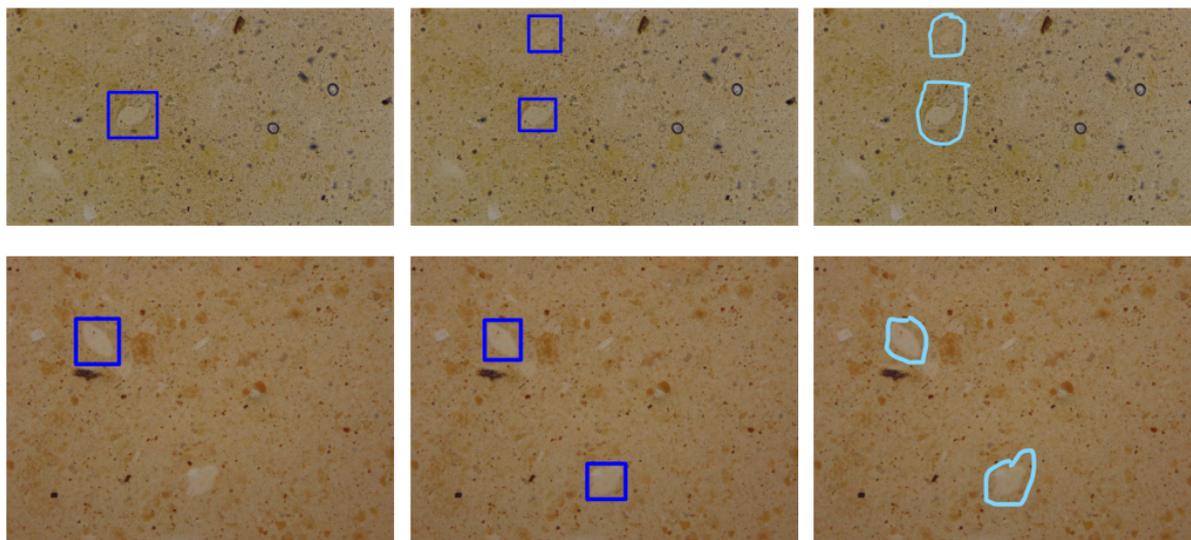


Figura 50 – Anotações realizadas pelos especialistas A, B e C nas duas imagens em que os modelos de aprendizado de máquina classificaram incorretamente dois ovos da espécie *S. mansoni*.

real para o diagnóstico da esquistossomose.

Em contrapartida, a Figura 51 apresenta alguns exemplos de casos em que o uso dos modelos de aprendizado de máquina contribuiu positivamente para a diminuição do número de falsos positivos gerados pelo método *Faster* R-CNN com uma ResNet-50.

A primeira coluna da Figura 51 ilustra as imagens do conjunto de teste com suas anotações originais, enquanto a segunda coluna apresenta as detecções realizadas pelo *Faster* R-CNN sozinho e a terceira coluna com a solução proposta nesse estudo integrada com os modelos de aprendizado de máquina. Em todos os casos apresentados, o sistema de diagnóstico de esquistossomose conseguiu detectar corretamente todos os ovos da espécie *S. mansoni*, sem que fosse gerado nenhum falso positivo.

A solução proposta também foi testada com imagens que não continham nenhum ovo da espécie *S. mansoni*, com o objetivo de verificar se seria gerado falsos positivos. A Figura 52 mostra exemplos de detecções incorretas de ovos de *S. mansoni* que foram removidos pelos métodos de aprendizado de máquina por se tratar de falsos positivos.

Assim como na Figura 51, a primeira coluna da Figura 52 mostra as imagens originais avaliadas pelos especialistas. Já a segunda coluna apresenta as detecções realizadas pelo *Faster* R-CNN sozinho e a terceira coluna o modelo de aprendizado profundo integrado com os modelos de aprendizado de máquina. Como pode ser visto na Figura 52, caso os

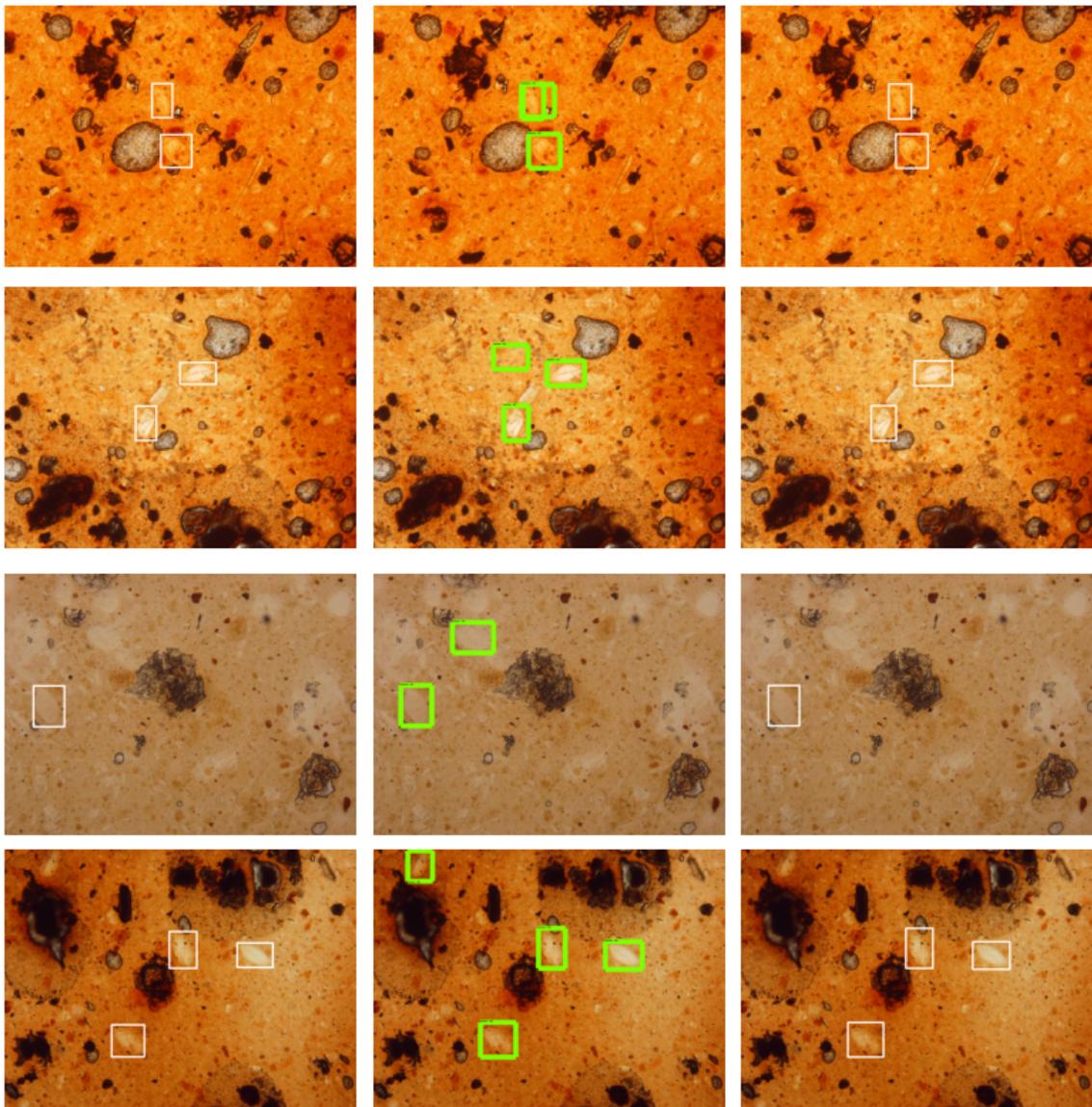


Figura 51 – Exemplos de detecção em que a integração do método *Faster R-CNN* com os modelos de aprendizado de máquina ajudaram a amenizar as detecções incorretas.

modelos de aprendizado de máquina não tivessem sido integrados a solução desenvolvida, todas as caixas delimitadoras ilustradas pela cor verde seriam apresentadas ao especialista. Graças a integração, as detecções realizadas pelo método de detecção de objetos *Faster R-CNN* com uma *ResNet-50* foram descartadas uma vez que foram consideradas como falsos positivos.

Com os experimentos realizados se percebeu que, quando foi utilizado um limiar de detecção de 0.1 para o método de detecção de objetos *Faster R-CNN* e um limite de confiança de 0.5 para os modelos de aprendizado de máquina *Random Forest*, *SVM* e *XGBoost*, conseguiu-se detectar os ovos que não haviam sido anteriormente detectados

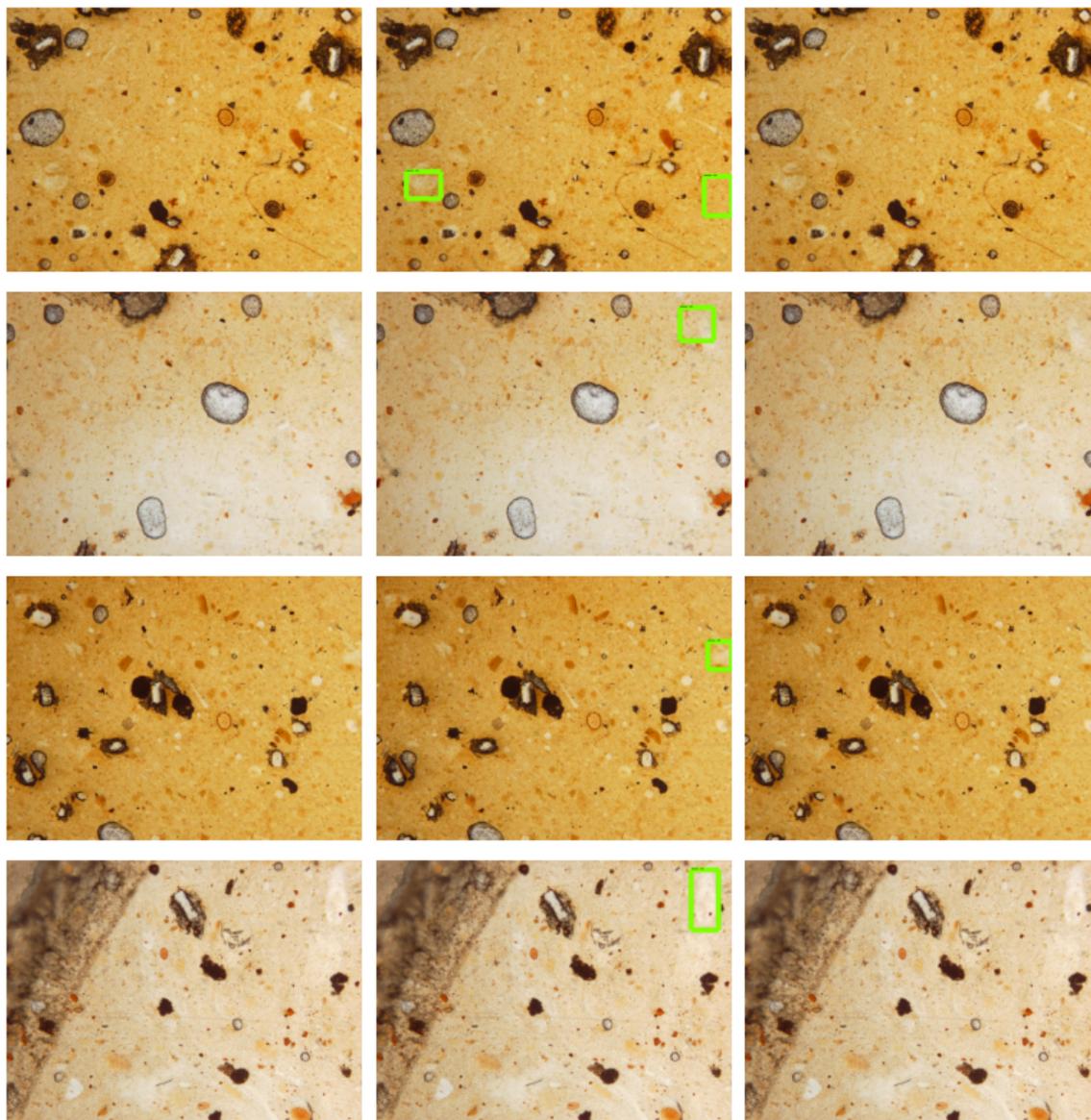


Figura 52 – Exemplos de imagens sem ovos de *S. mansoni* e que a integração do método *Faster R-CNN* com os modelos de aprendizado de máquina ajudaram a amenizar a quantidade de falsos positivos.

com um limiar superior. Porém, quando testado com valores alternativos, alguns fungos começavam a ser detectados, gerando o falso positivo e ocasionando detecções incorretas.

Dessa forma, foi considerado que é mais viável obter falsos positivos (detectar sujeira como ovo), do que não detectar um ovo que deveria ter sido detectado. Além disso, ressalta-se que a solução proposta não deve ser usada como diagnóstico único e final da esquistossomose, mas sim um apoio à tomada de decisão do profissional da saúde, que poderá validar ou não os resultados após o uso do sistema.

Ressalta-se que, como discutido anteriormente, para o conjunto de teste adotado

o desempenho da solução muda de acordo com os limiares de detecção e com os limites de confiança explorados. Baseado no estudo realizado para essas imagens de teste, foi visto que a melhor configuração de calibração é utilizar um limiar de detecção de 0.1 para o método de detecção de objetos *Faster R-CNN* e um limite de confiança de 0.5 para os modelos de aprendizado de máquina *Random Forest*, *SVM* e *XGBoost*. Contudo, sabe-se que o procedimento ideal é de definir esses parâmetros de calibração para a inferência com base em um conjunto de validação para que, posteriormente, testasse a melhor solução encontrada em outros dados de teste.

6.6 Comparativo com imagens dos trabalhos relacionados

Como descrito no capítulo de Material e Métodos em relação ao comparativo com os trabalhos relacionados, a Tabela 12 apresenta o resultado médio dos cinco experimentos realizados com imagens de lâminas que não foram preparadas pela técnica parasitológica Kato-Katz. Já a Figura 53 ilustra exemplos de detecções em imagens de teste realizadas pela solução aqui proposta, quando treinada com os *datasets* empregados nos trabalhos relacionados.

Tabela 12 – Resultados obtidos com as imagens utilizadas nos trabalhos relacionadas.

Métrica	IoU	área	maxDets	Resultado
AP	0.50:0.95	all	100	0.910
	0.50			1.000
	0.75			1.000
	0.50:0.95	small		-1
		medium		-1
		large		0.910
AR		all	1	0.928
			10	0.928
	100		0.928	
		small	-1	
		medium	-1	
large	0.928			

Quando são comparados os resultados obtidos com os trabalhos relacionados, não é possível considerar como sendo uma comparação justa e viável. O método utilizado nessa pesquisa para preparação das lâminas contendo amostra fecal foi a técnica de Kato-Katz,

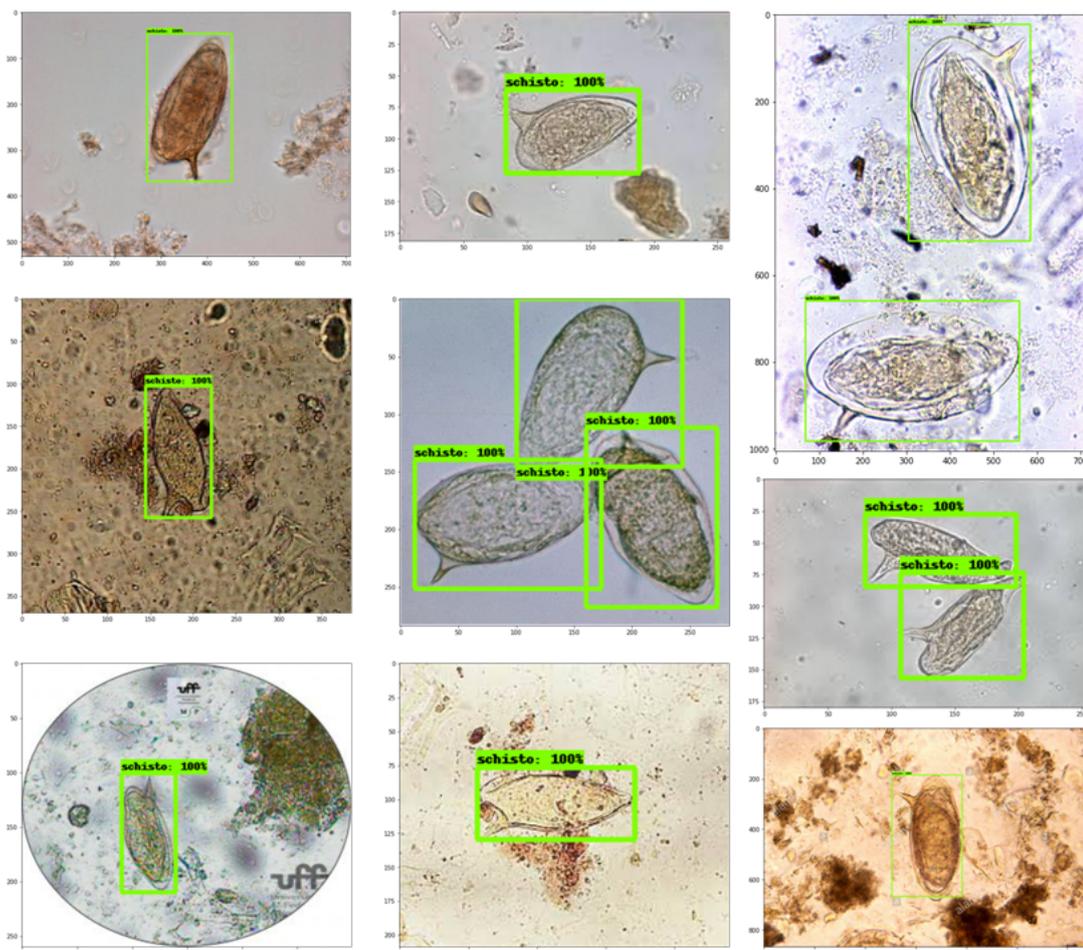


Figura 53 – Detecções realizadas pela solução apresentada nesse estudo em imagens utilizadas nos trabalhos relacionados.

que é eficiente na detecção de infecções moderadas e intensas; entretanto, indivíduos com baixas cargas parasitárias não são facilmente identificadas por este método (De Vlas and Gryseels, 1992; Oliveira et al., 2018). Trabalhos relacionados utilizam técnicas alternativas (muitos deles nem citam o nome do método empregado, porém, pelas imagens apresentadas é possível estimar a técnica com o auxílio de um especialista), fazendo com que a lâmina seja mais limpa e o ovo fique evidente na imagem. Em alguns casos, as imagens que foram utilizadas nem sequer fazem sentido, quando é considerado um cenário de aplicação em ambiente real.

Além disso, parte dos trabalhos relacionados aplicam ao microscópio um *zoom* muito maior do que o empregado nesse estudo, gerando algumas implicações. Com um *zoom* muito alto, consequentemente os ovos são mais representativos na imagem, tornando a detecção mais fácil. Em contrapartida, seria necessário um número muito maior de imagens para cobrir toda a lâmina contendo amostra fecal, fazendo com que o especialista

gaste muito mais tempo na captura das imagens, inviabilizando o uso da solução.

6.7 Comparativo com a ferramenta ParasiteID

Por fim, para o comparativo com a ferramenta ParasiteID, 30 imagens originais contendo ovos da espécie *S. mansoni* foram escolhidas aleatoriamente e submetidas à ferramenta, para que fossem analisadas individualmente. Em 28 casos, a ferramenta retornou a seguinte mensagem como resposta: “Image does NOT contain any parasite eggs.”. Em duas imagens, a resposta foi: “Sorry, we can’t make a good prediction for this image. Please try uploading a clearer picture.”. A Figura 54 apresenta as duas imagens em que a ferramenta ParasiteID detectou um objeto.

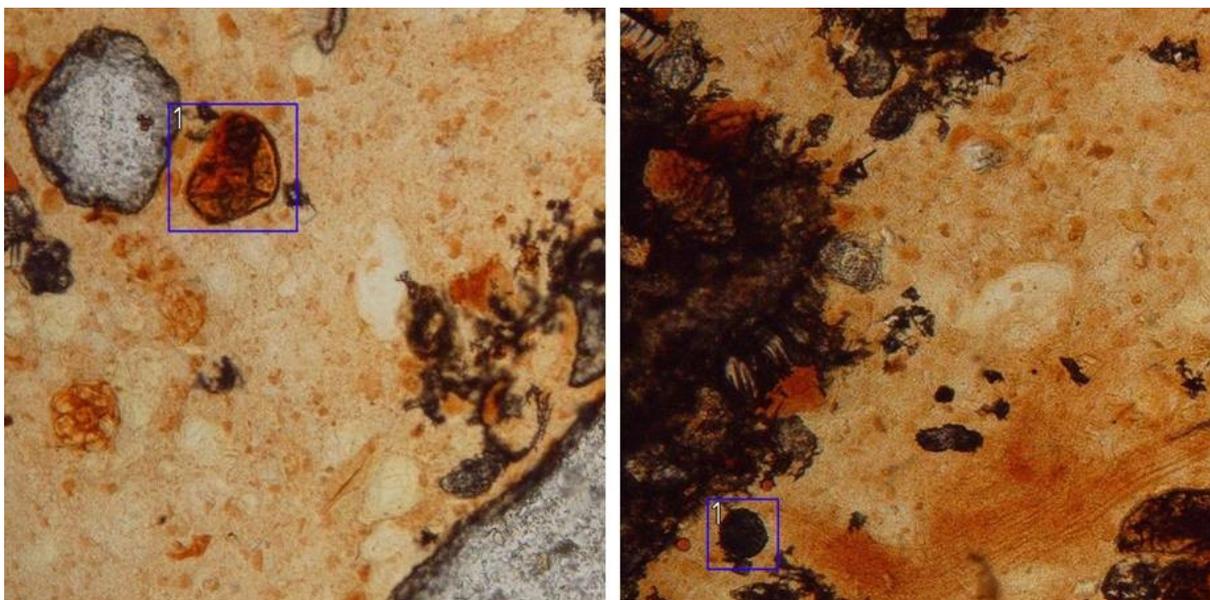


Figura 54 – Imagens detectadas pela ferramenta ParasiteID.

Conforme foi possível observar nos resultados obtidos com as diversas técnicas propostas, a tarefa de detecção de ovos da espécie *S. mansoni* não é nada trivial, até mesmo para métodos de última geração do aprendizado profundo. Além disso, os experimentos realizados com a ferramenta ParasiteID corroboram tal dificuldade. Apesar da eficiência de mais de 99% sugerida pelos desenvolvedores, a técnica se mostrou ineficiente com lâminas preparadas pelo método Kato-Katz, sendo que, das 30 amostras, nenhum ovo da espécie *S. mansoni* foi identificado corretamente, gerando o resultado de falso negativo.

Assim como apresentado na Figura 54, além dos ovos da espécie *S. mansoni* não terem sido detectados corretamente, a ferramenta ParasiteID ainda gerou falsos positivos,

confundindo sujeira com ovo de parasito. Apesar das limitações do estudo proposto neste trabalho, fica evidente seu uso no auxílio a tomada de decisão dos especialistas no diagnóstico da esquistossomose, sendo uma alternativa promissora para a área médica.

Após os experimentos realizados com as imagens de lâminas que não foram preparadas pela técnica parasitológica Kato-Katz, ficou ainda mais evidente a dificuldade de se trabalhar com essa técnica específica. Os trabalhos relacionados empregam técnicas parasitológicas alternativas, inviabilizando uma aplicação prática das soluções propostas.

A solução conseguiu generalizar muito bem para as imagens dos *datasets* alternativos, alcançando um AP de 100%, quando considerado a IoU como 0.50. Foi observada durante os experimentos uma rápida convergência do treinamento, visto que a natureza do problema pode ser considerada fácil (quando não se é empregada a técnica de preparação parasitológica Kato-Katz) para métodos de detecção de objetos de última geração, que é o caso desse estudo.

6.8 API

A Figura 55 apresenta a tela inicial do sistema proposto.

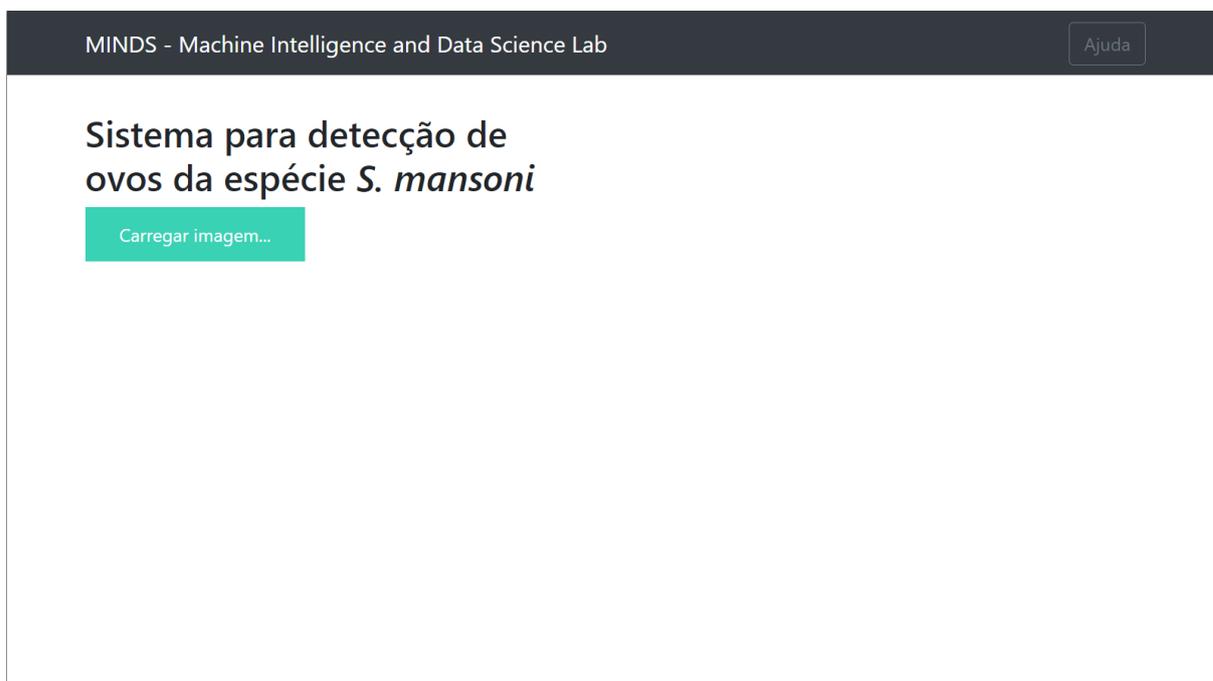


Figura 55 – Tela inicial do sistema proposto.

Como pode ser observado na Figura 55, a interface gráfica foi implementada para

ser simples e intuitiva, com o objetivo de ser de fácil entendimento e uso pelo profissional da área da saúde. Quando iniciado em um navegador de internet, o usuário clica no botão “Carregar imagem...” e consegue selecionar o arquivo para *upload*. A Figura 56 ilustra o momento em que o usuário faz a escolha de uma imagem para diagnóstico e a ferramenta faz seu carregamento.

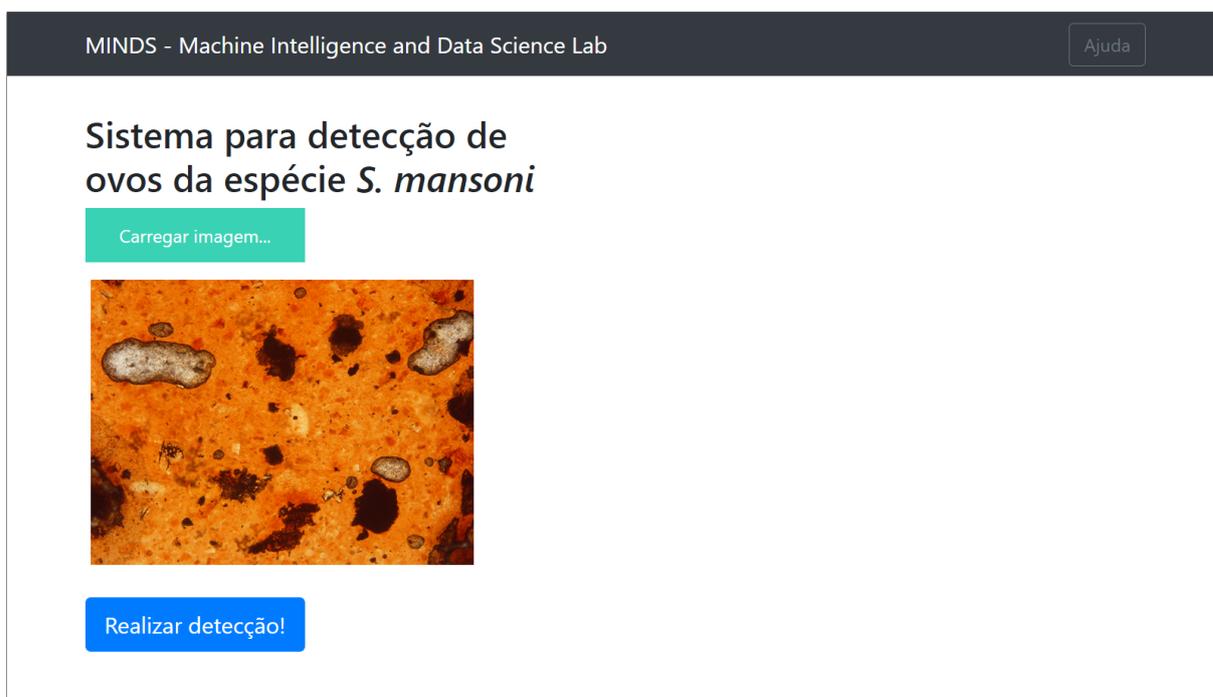


Figura 56 – Tela inicial após o usuário selecionar uma imagem.

Em seguida, após a imagem selecionada pelo usuário ter sido carregada pelo sistema, aparece na interface gráfica um botão “Realizar detecção!” com a opção de o usuário clicar sobre o mesmo e receber os resultados obtidos pelo modelo de inteligência artificial. A Figura 57 mostra as detecções realizadas da imagem carregada pelo usuário ilustrada na Figura 56.

Dessa forma, como apresentado na Figura 57, o usuário consegue visualizar embaixo da imagem uma mensagem indicando que o modelo de inteligência artificial detectou um ou mais ovos da espécie *S. Mansoni*. Além disso, sobre a imagem original apresentada na interface gráfica, o sistema desenvolvido desenha as caixas delimitadoras na imagem, indicando ao usuário o local exato de onde os ovos de *S. Mansoni* se encontram.

Por fim, caso o usuário escolha uma imagem que o modelo de inteligência artificial não consiga detectar nenhum ovo da espécie *S. Mansoni*, o resultado de diagnóstico do sistema é que nenhum ovo foi encontrado, mostrando essa mesma mensagem e não



Figura 57 – Tela da aplicação que apresenta o resultado de um caso em que havia ovos da espécie *S. Mansoni* na imagem.

desenhando nenhuma caixa delimitadora sobre a imagem original. A Figura 58 ilustra esse caso.



Figura 58 – Tela da aplicação que apresenta o resultado de um caso em que não foi detectado nenhum ovo da espécie *S. Mansoni* na imagem.

6.9 **Resumo**

Neste capítulo foram apresentados os resultados e discussões dos experimentos realizados com o sistema proposto. No próximo capítulo, são descritas as conclusões encontradas após desenvolvimento desta pesquisa, bem como os trabalhos futuros que podem vir a ser desenvolvidos a partir dessas considerações finais.

Capítulo 7

Conclusões

Nessa pesquisa de Doutorado foi proposta uma solução para diagnóstico automático da esquistossomose, por meio de imagens microscópicas de lâminas com material fecal preparadas pela técnica Kato-Katz. Foi realizado um estudo com três diferentes métodos de detecção de objetos (*Faster R-CNN*, *SSD* e *YOLO*), variando suas arquiteturas de CNN. O objetivo do estudo foi, a partir de imagens coletadas por diferentes microscópios, efetuar o treinamento de diferentes modelos de aprendizado profundo para que, em seguida, fosse possível definir qual seria o melhor método de detecção de objetos para fazer parte da solução proposta.

Para que o sistema conseguisse ser o mais acurado possível, foram treinados e testados os seguintes métodos de aprendizado de máquina: *K - Nearest Neighbors*, *Naive Bayes*, *Random Forest*, *Support Vector Machine* e *XGBoost*. Com isso, foi possível integrar os melhores modelos para auxiliar o método de detecção de objetos baseado em aprendizado profundo para a redução da quantidade de falsos positivos gerados pelo sistema.

Diferentemente dos trabalhos relacionados que não utilizaram imagens microscópicas que foram preparadas pela técnica parasitológica Kato-Katz, na solução proposta ela foi escolhida com a justificativa de ser o método mais comumente utilizado, prático e barato para o diagnóstico da esquistossomose. A partir do momento que decidiu-se utilizar tal metodologia, esbarrou-se com as dificuldades provindas por essa natureza de imagem, discutidas no decorrer do estudo. Apesar disso, manteve-se o foco que é de desenvolver uma ferramenta de inteligência artificial que seja realmente empregada e utilizada na prática, auxiliando os especialistas no diagnóstico da doença.

Conforme esperado, conclui-se que métodos de detecção de objetos *Faster R-CNN* conseguem ser mais acurados em suas detecções, em relação a outros detectores como SSD e YOLO. Contudo, o método YOLO tem a vantagem de conseguir realizar as detecções consideravelmente mais rápidas, em relação a seus principais concorrentes. O método de detecção de objetos SDD encontra-se no meio termo, teoricamente não possuindo um valor de AP tão bom quanto do *Faster R-CNN* nem tão baixo quanto do YOLO, além disso, não é tão rápido para realizar as detecções quanto o YOLO nem tão demorado quanto o *Faster R-CNN*.

Como a solução não demanda detecções em tempo real, optou-se pela escolha do *Faster R-CNN*, uma vez que ele apresentou o melhor valor para a métrica AP. Quando analisado a AP @[IoU=0.50], verificou-se que foi obtido um valor de 0.884, interessante para a detecção dos ovos da espécie de *S. mansoni* em lâminas preparadas pela técnica Kato-Katz, em que foi visto o quão dificultoso é realizar esse tipo de tarefa. Além disso, a integração com modelos de aprendizado de máquina auxiliou na redução da quantidade de falsos positivos, uma vez que foi preferido aumentar a sensibilidade do método de detecção *Faster R-CNN*, diminuindo o valor do limiar de detecção.

A solução desenvolvida nessa pesquisa foi empregada em imagens de lâminas que não foram preparadas pela técnica parasitológica Kato-Katz, podendo assim ser comparada com os trabalhos relacionados. Como resultado, obteve-se um valor de 100, considerando o AP @[IoU=0.50], indicando que a solução aqui proposta foi preparada para lidar com um problema de natureza mais complexa em relação do que foi apresentado na literatura.

Além disso, para corroborar ainda mais com a dificuldade de detecção de ovos da espécie *S. mansoni* em imagens microscópicas preparadas pela técnica Kato-Katz, viu-se que, ao se realizar a inferência das imagens em um aplicativo comercial, a ferramenta não conseguiu retornar nenhuma detecção correta. Sendo assim, pôde-se concluir o quanto é difícil desenvolver uma solução que possa ser aplicada na prática, extrapolando os limites de uma pesquisa em um ambiente controlado. Essa dificuldade também foi percebida com as marcações dos especialistas nas imagens, uma vez que até mesmo profissionais treinados divergem opiniões em relação a alguns ovos da espécie *S. mansoni*.

Como trabalhos futuros, outros métodos de detecção de objetos baseados em aprendizado profundo podem ser empregados para diagnóstico da esquistossomose por

meio de imagens microscópicas de lâminas de amostra fecal preparadas pela técnica parasitológica Kato-Katz, tais como: CenterNet (Duan et al., 2019), EfficientDet (Tan et al., 2020), ExtremeNet (Zhou et al., 2019), RefineNet (Lin et al., 2017a) e RetinaNet (Lin et al., 2017b), além de abordagens baseadas processamento digital de imagens clássico e aprendizado de máquina. Esses últimos exigem bem menos processamento, sendo possível primeiramente fazer um uso de um descritor de recursos diferente do HOG, como o SIFT ou SURF para que, em seguida, os recursos obtidos sejam submetidos a um modelo de classificação diferente dos que foram avaliados nessa pesquisa. Além disso, coletar mais imagens com diferentes modelos de microscópicos para que seja possível testar a capacidade de generalização do modelo desenvolvido e conseqüentemente seus parâmetros de calibração.

O estudo se mostrou uma alternativa para o diagnóstico automático da esquistossomose, por meio da detecção de ovos da espécie *S. mansoni* em imagens microscópicas. Novas espécies de parasitos podem ser acopladas a solução, para que seja possível um único sistema que consiga identificar diferentes classes de helmintos. Dessa forma, o sistema pode ser aplicado no Sistema Único de Saúde (SUS), auxiliando a tomada de decisão dos profissionais de saúde e agilizando o processo de diagnóstico de pacientes com suspeita de esquistossomose, além de poder ser empregado em áreas endêmicas e descobrir casos de pessoas que não possuem nenhuma suspeita da doença.

Referências

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.
- Akselrod-Ballin, A., Karlinsky, L., Hazan, A., Bakalo, R., Horesh, A. B., Shoshan, Y., and Barkan, E. (2019). Deep learning for automatic detection of abnormal findings in breast mammography. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 321–329. Springer.
- Al-Thanoon, N. A., Qasim, O. S., and Algamal, Z. Y. (2019). A new hybrid firefly algorithm and particle swarm optimization for tuning parameter estimation in penalized support vector machine with application in chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 184:142–152.
- Alpaydin, E. (2014). Introduction to machine learning/ethem alpaydin.
- Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.
- Ash, L. R., Orihel, T. C., and Savioli, L. (1994). *Bench aids for the diagnosis of intestinal parasites*. Organización Mundial de la Salud.
- Avcı, D. and Varol, A. (2009). An expert diagnosis system for classification of human parasite eggs based on multi-class svm. *Expert Systems with Applications*, 36(1):43–48.
- Barbosa, C. S., Gomes, E. C. S., Marcelino, J. M. R., Cavalcante, K. R., and Nascimento, W. R. C. (2017). Quality control of the slides by kato-katz method for the parasitological diagnosis of schistosomiasis infection by *Schistosoma mansoni*. *Jornal Brasileiro de Patologia e Medicina Laboratorial*, 53:110–114.

- Barreto, J. M. (2002). Introdução às redes neurais artificiais. *Florianópolis: UFSC*.
- Batista, G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29.
- Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer.
- Beam, A. L., Manrai, A. K., and Ghassemi, M. (2020). Challenges to the reproducibility of machine learning models in health care. *Jama*, 323(4):305–306.
- Belgiu, M. and Drăguț, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, 114:24–31.
- Bengio, Y., Goodfellow, I., and Courville, A. (2017). *Deep learning*, volume 1. Citeseer.
- Berhe, N., Medhin, G., Erko, B., Smith, T., Gedamu, S., Bereded, D., Moore, R., Habte, E., Redda, A., Gebre-Michael, T., et al. (2004). Variations in helminth faecal egg counts in kato–katz thick smears and their implications in assessing infection status with schistosoma mansoni. *Acta tropica*, 92(3):205–212.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification and regression trees*. Routledge.
- Butploy, N., Kanarkard, W., and Maleewong Intapan, P. (2021). Deep learning approach for ascaris lumbricoides parasite egg classification. *Journal of Parasitology Research*, 2021.
- Byvatov, E. and Schneider, G. (2003). Support vector machine applications in bioinformatics. *Applied bioinformatics*, 2(2):67–77.

- Cao, X., Wu, C., Yan, P., and Li, X. (2011). Linear svm classification using boosting hog features for vehicle detection in low-altitude airborne videos. In *2011 18th IEEE International Conference on Image Processing*, pages 2421–2424. IEEE.
- Carvalho, D. V., Pereira, E. M., and Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832.
- Carvalho, O. d. S., Mendonça, C. L. F. d., Marcelino, J. M. d. R., Passos, L. K. J., Fernandez, M. A., Leal, R. d. S., Caldeira, R. L., Scholte, R. G. C., Carmo, E. H., Mesquita, S. G., et al. (2018). Geographical distribution of intermediate hosts of schistosoma mansoni in the states of paran , minas gerais, bahia, pernambuco and rio grande do norte, brazil, 2012-2014. *Epidemiologia e Servi os de Sa de*, 27:e2017343.
- Castoril, B. R., Oliveira, B. A. S., Assis, S. S., Morais, B. R., and Santos, F. H. W. (2018). Previs o de demanda de energia el trica no setor de produ o do latic nio do ifmg campus bambu  utilizando regress o de vetores de suporte. *Revista de Inform tica Aplicada*, 14(2).
- Cavalcanti, M. G., Silva, L. F., Peralta, R. H., Barreto, M. G., and Peralta, J. M. (2013). Schistosomiasis in areas of low endemicity: a new era in diagnosis. *Trends in parasitology*, 29(2):75–82.
- Chai, J., Zeng, H., Li, A., and Ngai, E. W. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6:100134.
- Chang, C. C. and Lin, C. J. (2011). Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27.
- Chauhan, N. K. and Singh, K. (2018). A review on conventional machine learning vs deep learning. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 347–352. IEEE.
- Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6.

- Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.
- Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., et al. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4.
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.
- Chuah, C., Gobert, G. N., Latif, B., Heo, C. C., and Leow, C. Y. (2019). Schistosomiasis in malaysia: A review. *Acta tropica*, 190:137–143.
- Coelho, P. M. Z., Siqueira, L. M. V., Grenfell, R. F. Q., Almeida, N. B. F., Katz, N., Almeida, Á., Carneiro, N. F. d. F., and Oliveira, E. (2016). Improvement of poc-cca interpretation by using lyophilization of urine from patients with schistosoma mansoni low worm burden: towards an elimination of doubts about the concept of trace. *PLoS neglected tropical diseases*, 10(6):e0004778.
- Colley, D. G., Bustinduy, A. L., Secor, W. E., and King, C. H. (2014). Human schistosomiasis. *The Lancet*, 383(9936):2253–2264.
- Cristianini, N., Shawe-Taylor, J., et al. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.
- Dacal, E., Bermejo-Peláez, D., Lin, L., Álamo, E., Cuadrado, D., Martínez, Á., Mousa, A., Postigo, M., Soto, A., Sukosd, E., et al. (2021). Mobile microscopy and telemedicine platform assisted by deep learning for quantification of trichuris trichiura infection. *bioRxiv*.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE.
- De Carli, G. A. (2001). Parasitologia clínica: seleção de métodos e técnicas de laboratório para diagnóstico das parasitoses humanas. In *Parasitologia clínica: seleção de métodos*

- e técnicas de laboratório para diagnóstico das parasitoses humanas*, page 810. Atheneu: São Paulo.
- De Vlas, S. and Gryseels, B. (1992). Underestimation of schistosoma mansoni prevalences. *Parasitology Today*, 8(8):274–277.
- Deka, P. C. et al. (2014). Support vector machine applications in the field of hydrology: a review. *Applied soft computing*, 19:372–386.
- Delgado Huaynalaya, E. and Hirata, N. S. T. (2012). *Detecção de ovos de S. mansoni a partir da detecção de seus contornos*. PhD thesis, Universidade de São Paulo.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- Deng, L., Yu, D., et al. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387.
- Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.
- Ding, J., Li, A., Hu, Z., and Wang, L. (2017). Accurate pulmonary nodule detection in computed tomography images using deep convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 559–567. Springer.
- Dogantekin, E., Yilmaz, M., Dogantekin, A., Avci, E., and Sengur, A. (2008). A robust technique based on invariant moments–anfis for recognition of human parasite eggs in microscopic images. *Expert Systems with Applications*, 35(3):728–738.
- Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578.
- Edmonds, J. (1971). Matroids and the greedy algorithm. *Mathematical programming*, 1(1):127–136.

- Enk, M. J., Lima, A. C. L., Drummond, S. C., Schall, V. T., and Coelho, P. M. Z. (2008). The effect of the number of stool samples on the observed prevalence and the infection intensity with *Schistosoma mansoni* among a population in an area of low transmission. *Acta tropica*, 108(2-3):222–228.
- Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660.
- Espírito-Santo, M. C. C., Alvarado-Mora, M. V., Dias-Neto, E., Botelho-Lima, L. S., Moreira, J. P., Amorim, M., Pinto, P. L. S., Heath, A. R., Castilho, V. L. P., do Nascimento Gonçalves, E. M., et al. (2014). Evaluation of real-time pcr assay to detect schistosoma mansoni infections in a low endemic setting. *BMC Infectious Diseases*, 14(1):558.
- Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181.
- Fix, E. and Hodges, J. L. (1989). Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247.
- Flores, J. H. F. (2009). Comparação de modelos mlp/rna e modelos box-jenkins em séries temporais não lineares. *Porto Alegre: UFRGS*.
- Fuss, A., Mazigo, H. D., Tappe, D., Kasang, C., and Mueller, A. (2018). Comparison of sensitivity and specificity of three diagnostic tests to detect schistosoma mansoni infections in school children in mwanza region, tanzania. *PloS one*, 13(8):e0202499.
- Gandhi, R. (2018). R-cnn, fast r-cnn, faster r-cnn, yolo — object detection algorithms. <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>. Acesso em: 07/08/2021.
- Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4):42–47.

- Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.
- Goldstein, B. A., Hubbard, A. E., Cutler, A., and Barcellos, L. F. (2010). An application of random forests to a genome-wide association dataset: methodological considerations & new findings. *BMC genetics*, 11(1):1–13.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.
- Grinberg, M. (2018). *Flask web development: developing web applications with python*. "O'Reilly Media, Inc."
- Gryseels, B., Polman, K., Clerinx, J., and Kestens, L. (2006). Human schistosomiasis. *The Lancet*, 368(9541):1106–1118.
- Hagan, M. T., Demuth, H. B., and Beale, M. (1997). *Neural network design*. PWS Publishing Co.
- Hart, P. (1968). The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 14(3):515–516.
- Hay, A. (1988). The derivation of global estimates from a confusion matrix. *International Journal of Remote Sensing*, 9(8):1395–1398.
- He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284.
- He, H. and Ma, Y. (2013). Imbalanced learning: foundations, algorithms, and applications. *Imbalanced Learning: Foundations, Algorithms, and Applications*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier.

- Hinz, R., Schwarz, N. G., Hahn, A., and Frickmann, H. (2017). Serological approaches for the diagnosis of schistosomiasis—a review. *Molecular and Cellular Probes*, 31:2–21.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017a). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.
- Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al. (2017b). Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311.
- Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of physiology*, 160(1):106–154.
- Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. *arXiv preprint arXiv:1602.07360*.
- Katz, N. (2018). Inquérito nacional de prevalência da esquistossomose mansoni e geohelminthoses.
- Katz, N., Chaves, A., Pellegrino, J., et al. (1972). A simple device for quantitative stool thick-smear technique in schistosomiasis mansoni. *Rev Inst Med Trop Sao Paulo*, 14(6):397–400.
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., Corlay, S., et al. (2016). Jupyter notebooks—a publishing format for reproducible computational workflows. In *ELPUB*, pages 87–90.
- Kong, X., Meng, Z., Nojiri, N., Iwahori, Y., Meng, L., and Tomiyama, H. (2019). A hog-svm based fall detection iot system for elderly persons using deep sensor. *Procedia computer science*, 147:276–282.

- Kongs, A., Marks, G., Verle, P., and Van Der Stuyft, P. (2001). The unreliability of the kato-katz technique limits its usefulness for evaluating *s. mansoni* infections. *Tropical Medicine & International Health*, 6(3):163–169.
- Koyejo, O., Natarajan, N., Ravikumar, P., and Dhillon, I. S. (2014). Consistent binary classification with generalized performance metrics. In *NIPS*, volume 27, pages 2744–2752. Citeseer.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.
- Kubat, M., Matwin, S., et al. (1997). Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, volume 97, pages 179–186. Citeseer.
- Larsson, J. and Hedberg, R. (2020). Development of machine learning models for object identification of parasite eggs using microscopy.
- LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.
- LeCun, Y., Kavukcuoglu, K., and Farabet, C. (2010). Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 253–256. IEEE.
- Lee, S.-H., Bang, M., Jung, K.-H., and Yi, K. (2015). An efficient selection of hog feature for svm classification of vehicle. In *2015 International Symposium on Consumer Electronics (ISCE)*, pages 1–2. IEEE.
- Li, F.-F., Johnson, J., and Yeung, S. (2017). Recurrent neural networks.
- Lim, S. and Chi, S. (2019). Xgboost application on bridge management systems for proactive damage estimation. *Advanced Engineering Informatics*, 41:100922.

- Lin, G., Milan, A., Shen, C., and Reid, I. (2017a). Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017b). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.
- Lowd, D. and Domingos, P. (2005). Naive bayes models for probability estimation. In *Proceedings of the 22nd international conference on Machine learning*, pages 529–536.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.
- Mahmood, T., Arsalan, M., Owais, M., Lee, M. B., and Park, K. R. (2020). Artificial intelligence-based mitosis detection in breast cancer histopathology images using faster r-cnn and deep cnns. *Journal of Clinical Medicine*, 9(3):749.
- Martins-Melo, F. R., Carneiro, M., Ramos Jr, A. N., Heukelbach, J., Ribeiro, A. L. P., and Werneck, G. L. (2018). The burden of neglected tropical diseases in brazil, 1990-2016: a subnational analysis from the global burden of disease study 2016. *PLoS neglected tropical diseases*, 12(6):e0006559.
- Michie, D., Spiegelhalter, D. J., Taylor, C., et al. (1994). Machine learning. *Neural and Statistical Classification*, 13.
- Moses and Olafenwa, J. (2018). Imageai, an open source python library built to empower developers to build applications and systems with self-contained computer vision capabilities.

- Naghibi, S. A., Ahmadi, K., and Daneshi, A. (2017). Application of support vector machine, random forest, and genetic algorithm optimized random forest models in groundwater potential mapping. *Water Resources Management*, 31(9):2761–2775.
- Naidoo, P., Ghazi, T., Chuturgoon, A. A., Naidoo, R. N., Ramsuran, V., Mpaka-Mbatha, M. N., Bhengu, K. N., Nembe, N., Duma, Z., Pillay, R., et al. (2021). Sars-cov-2 and helminth co-infections, and environmental pollution exposure: An epidemiological and immunological perspective. *Environment International*, page 106695.
- Nascimento, G. L., Pegado, H. M., Domingues, A. L. C., Ximenes, R. A. d. A., Itria, A., Cruz, L. N., and Oliveira, M. R. F. d. (2019). The cost of a disease targeted for elimination in brazil: the case of schistosomiasis mansoni. *Memórias do Instituto Oswaldo Cruz*, 114.
- Neves, D. P. (2003). Parasitologia dinâmica. In *Parasitologia dinâmica*, page 212. Atheneu: São Paulo.
- Nkamgang, O. T., Tchiotsop, D., Fotsin, H. B., Talla, P. K., Dorr, V. L., and Wolf, D. (2019). Automating the clinical stools exam using image processing integrated in an expert system. *Informatics in Medicine Unlocked*, 15:100165.
- Nkamgang, O. T., Tchiotsop, D., Tchinda, B. S., and Fotsin, H. B. (2018). A neuro-fuzzy system for automated detection and classification of human intestinal parasites. *Informatics in Medicine Unlocked*, 13:81–91.
- Noble, W. S. et al. (2004). Support vector machine applications in computational biology. *Kernel methods in computational biology*, 14:71–92.
- Nunes, G. H., Oliveira, B. A. S., and Nametala, C. A. L. (2020). A computational strategy for classification of enem issues based on item response theory. *Revista de Informática Teórica e Aplicada*, 27(1):92–111.
- Oliveira, W. J., Magalhães, F. d. C., Elias, A. M. S., de Castro, V. N., Favero, V., Lindholz, C. G., Oliveira, Á. A., Barbosa, F. S., Gil, F., Gomes, M. A., et al. (2018). Evaluation of diagnostic methods for the detection of intestinal schistosomiasis in endemic areas with low parasite loads: Saline gradient, helmintex, kato-katz and rapid urine test. *PLoS neglected tropical diseases*, 12(2):e0006232.

- Padilla, R., Netto, S. L., and da Silva, E. A. B. (2020). A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242.
- Pan, B. (2018). Application of xgboost algorithm in hourly pm_{2.5} concentration prediction. In *IOP conference series: Earth and environmental science*, volume 113, page 012127. IOP publishing.
- Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.
- Parsa, A. B., Movahedi, A., Taghipour, H., Derrible, S., and Mohammadian, A. K. (2020). Toward safer highways, application of xgboost and shap for real-time accident detection and feature analysis. *Accident Analysis & Prevention*, 136:105405.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.
- Peñas, K. E., Villacorte, E. A., Rivera, P. T., and Naval, P. C. (2020). Automated detection of helminth eggs in stool samples using convolutional neural networks. In *2020 IEEE REGION 10 CONFERENCE (TENCON)*, pages 750–755. IEEE.
- Perez, L. and Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.
- Pontes, L. A., Dias-Neto, E., and Rabello, A. (2002). Detection by polymerase chain reaction of schistosoma mansoni dna in human serum and feces. *The American journal of tropical medicine and hygiene*, 66(2):157–162.
- Provost, F. (2000). Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI’2000 workshop on imbalanced data sets*, volume 68, pages 1–3. AAAI Press.
- Rabello, A., Pontes, L., Enk, M., Montenegro, S., and Morais, C. (2008). Diagnóstico parasitológico, imunológico e molecular da esquistossomose mansoni. *OS Carvalho, PMZ Coelho, HL Lenzi, Schistosoma mansoni & esquistossomose, Fiocruz, Rio de Janeiro*, pages 897–925.

- Redmon, J. (2013–2016). Darknet: Open source neural networks in c. <http://pjreddie.com/darknet/>.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- Redmon, J. and Farhadi, A. (2017). Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271.
- Redmon, J. and Farhadi, A. (2018). YoloV3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.
- Rish, I. et al. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46.
- Roberts, M., Cross, J., Pohl, U., Lucas, S., and Dean, A. (2006). Cerebral schistosomiasis. *The Lancet Infectious Diseases*, 6(12):820.
- Rosati, R., Romeo, L., Silvestri, S., Marcheggiani, F., Tiano, L., and Frontoni, E. (2020). Faster r-cnn approach for detection and quantification of dna damage in comet assay images. *Computers in Biology and Medicine*, 123:103912.
- Rosebrock, A. (2016). Intersection over union (iou) for object detection. <http://https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection>. Acesso em: 07/08/2021.
- Rosenberg, C., Hebert, M., and Schneiderman, H. (2005). Semi-supervised self-training of object detection models. *WACV/MOTION*, 2.
- Ross, A. G., Olveda, R. M., Chy, D., Olveda, D. U., Li, Y., Harn, D. A., Gray, D. J., McManus, D. P., Tallo, V., Chau, T. N., et al. (2015). Can mass drug administration lead to the sustainable control of schistosomiasis? *The Journal of infectious diseases*, 211(2):283–289.

- Sa, R., Owens, W., Wiegand, R., Studin, M., Capoferri, D., Barooha, K., Greaux, A., Rattray, R., Hutton, A., Cintineo, J., et al. (2017). Intervertebral disc detection in x-ray images using faster r-cnn. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 564–567. IEEE.
- Savioli, L., Albonico, M., Colley, D. G., Correa-Oliveira, R., Fenwick, A., Green, W., Kabatereine, N., Kabore, A., Katz, N., Klohe, K., et al. (2017). Building a global schistosomiasis alliance: an opportunity to join forces to fight inequality and rural poverty. *Infectious Diseases of Poverty*, 6(1):1–6.
- Saxena, P. (2021). Selective search for object detection | r-cnn. <https://www.geeksforgeeks.org/selective-search-for-object-detection-r-cnn/>. Acesso em: 07/08/2021.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.
- Shan, Q., Li, Z., Jia, J., and Tang, C.-K. (2008). Fast image/video upsampling. *ACM Transactions on Graphics (TOG)*, 27(5):1–7.
- Sheng, B., Zhou, M., Hu, M., Li, Q., Sun, L., and Wen, Y. (2020). A blood cell dataset for lymphoma classification using faster r-cnn. *Biotechnology & Biotechnological Equipment*, 34(1):413–420.
- Shinde, P. P. and Shah, S. (2018). A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBE)*, pages 1–6. IEEE.
- Silva-Moraes, V., Shollenberger, L. M., Siqueira, L. M. V., Castro-Borges, W., Harn, D. A., Rabello, A. L. T., Coelho, P. M. Z., et al. (2019). Diagnosis of schistosoma mansoni infections: what are the choices in brazilian low-endemic areas? *Memórias do Instituto Oswaldo Cruz*, 114.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Siqueira, L. M. V., Gomes, L. I., Oliveira, E., Oliveira, E. R. d., Oliveira, Á. A. d., Enk, M. J., Carneiro, N. F., Rabello, A., and Coelho, P. M. Z. (2015). Evaluation of

- parasitological and molecular techniques for the diagnosis and assessment of cure of schistosomiasis mansoni in a low transmission area. *Memórias do Instituto Oswaldo Cruz*, 110:209–214.
- Sokolova, M., Japkowicz, N., and Szpakowicz, S. (2006). Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, pages 1015–1021. Springer.
- Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, 62(1):77–89.
- Suzuki, C. T., Gomes, J. F., Falcao, A. X., Papa, J. P., and Hoshino-Shimizu, S. (2012). Automatic segmentation and classification of human intestinal parasites from microscopy images. *IEEE Transactions on biomedical Engineering*, 60(3):803–812.
- Suzuki, C. T., Gomes, J. F., Falcao, A. X., Shimizu, S. H., and Papa, J. P. (2013). Automated diagnosis of human intestinal parasites using optical microscopy images. In *2013 IEEE 10th International Symposium on Biomedical Imaging*, pages 460–463. IEEE.
- Syriopoulos, T., Tsatsaronis, M., and Karamanos, I. (2021). Support vector machine algorithms: An application to ship price forecasting. *Computational Economics*, 57(1):55–87.
- Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.
- Szegedy, C., Reed, S., Erhan, D., Anguelov, D., and Ioffe, S. (2014). Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441*.
- Tan, M., Pang, R., and Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790.
- Tarafder, M., Carabin, H., Joseph, L., Balolong Jr, E., Olveda, R., and McGarvey, S. (2010). Estimating the sensitivity and specificity of kato-katz stool examination technique for

- detection of hookworms, ascaris lumbricoides and trichuris trichiura infections in humans in the absence of a ‘gold standard’. *International journal for parasitology*, 40(4):399–404.
- Tchinda, B. S., Noubom, M., Tchiotsop, D., Louis-Dorr, V., and Wolf, D. (2019). Towards an automated medical diagnosis system for intestinal parasitosis. *Informatics in Medicine Unlocked*, page 100238.
- Ting, K. M., Sammut, C., and Webb, G. (2011). Encyclopedia of machine learning. *KM Ting.–Boston, MA: Springer*.
- Tomek, I. (1976). Two modifications of cnn. *I.E.E.E. Trans. Syst. Man cybern.; U.S.A.; da. 1976; vol. 6; no 11; pp. 769-772; bibl. 7 ref.*
- Tyagi, S. and Mittal, S. (2020). Sampling approaches for imbalanced data classification problem in machine learning. In *Proceedings of ICRIC 2019*, pages 209–221. Springer.
- Tzutalin, L. (2015). Git code (2015).
- Uçar, M. K., Nour, M., Sindi, H., and Polat, K. (2020). The effect of training and testing process on machine learning in biomedical datasets. *Mathematical Problems in Engineering*, 2020.
- Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, 104(2):154–171.
- Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.
- Viet, N. Q., ThanhTuyen, D. T., and Hoang, T. H. (2019). Parasite worm egg automatic detection in microscopy stool image based on faster r-cnn. In *Proceedings of the 3rd international conference on machine learning and soft computing*, pages 197–202.
- Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. IEEE.
- Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018.

- Vuttipittayamongkol, P., Elyan, E., and Petrovski, A. (2021). On the class overlap problem in imbalanced data classification. *Knowledge-based systems*, 212:106631.
- Wangenheim, A. (2019). Visão computacional::deep learning. <https://www.lapix.ufsc.br/ensino/visao/visao-computacionaldeep-learning/>. Acesso em: 07/08/2021.
- Weerakoon, K. G., Gordon, C. A., and McManus, D. P. (2018). Dna diagnostics for schistosomiasis control. *Tropical medicine and infectious disease*, 3(3):81.
- WHO (2020). *Bench aids for the diagnosis of intestinal parasites*. World Health Organization.
- WHO (2021). Schistosomiasis and soil-transmitted helminthiasis: progress report, 2020–schistosomiase et géohelminthiasis: rapport de situation, 2020. *Weekly Epidemiological Record= Relevé épidémiologique hebdomadaire*, 96(48):585–595.
- Wong, T.-T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, 48(9):2839–2846.
- Xing, W., Deng, N., Xin, B., Liu, Y., Chen, Y., and Zhang, Z. (2019). Identification of extremely similar animal fibers based on matched filter and hog-svm. *IEEE Access*, 7:98603–98617.
- Xu, L., Ren, J. S., Liu, C., and Jia, J. (2014). Deep convolutional neural network for image deconvolution. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.
- Yadav, R. P., Senthilarasu, V., Kutty, K., and Ugale, S. P. (2015). Implementation of robust hog-svm based pedestrian classification. *International Journal of Computer Applications*, 114(19).
- Yang, S., Fang, B., Tang, W., Wu, X., Qian, J., and Yang, W. (2018). Faster r-cnn based microscopic cell detection. In *2018 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, pages 345–350. IEEE.
- Yang, Y. S., Park, D. K., Kim, H. C., Choi, M.-H., and Chai, J.-Y. (2001). Automatic identification of human helminth eggs on microscopic fecal specimens using digital

- image processing and an artificial neural network. *IEEE Transactions on Biomedical Engineering*, 48(6):718–730.
- Yao, C., Wu, F., Chen, H.-j., Hao, X.-l., and Shen, Y. (2014). Traffic sign recognition using hog-svm and grid search. In *2014 12th International Conference on Signal Processing (ICSP)*, pages 962–965. IEEE.
- Zendehboudi, A., Baseer, M. A., and Saidur, R. (2018). Application of support vector machine models for forecasting solar and wind energy resources: A review. *Journal of cleaner production*, 199:272–285.
- Zhang, L. and Zhan, C. (2017). Machine learning in rock facies classification: an application of xgboost. In *International Geophysical Conference, Qingdao, China, 17-20 April 2017*, pages 1371–1374. Society of Exploration Geophysicists and Chinese Petroleum Society.
- Zhang, Z. (2016). Derivation of backpropagation in convolutional neural network (cnn). *University of Tennessee, Knoxville, TN*.
- Zhou, X., Zhuo, J., and Krahenbuhl, P. (2019). Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 850–859.
- Zoni, A. C., Catalá, L., and Ault, S. K. (2016). Schistosomiasis prevalence and intensity of infection in latin america and the caribbean countries, 1942-2014: a systematic review in the context of a regional elimination goal. *PLoS neglected tropical diseases*, 10(3):e0004493.