

**UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA**

TIAGO RAMOS DE SÁ

**EMBARCAÇÕES AUTÔNOMAS NA REGIÃO AMAZÔNICA:
APLICAÇÃO DE TÉCNICAS DE APRENDIZADO PROFUNDO PARA
DETECÇÃO DE OBJETOS LOCALIZADOS NA SUPERFÍCIE DE RIOS**

MANAUS

2022

TIAGO RAMOS DE SÁ

**EMBARCAÇÕES AUTÔNOMAS NA REGIÃO AMAZÔNICA:
APLICAÇÃO DE TÉCNICAS DE APRENDIZADO PROFUNDO PARA
DETECÇÃO DE OBJETOS LOCALIZADOS NA SUPERFÍCIE DE RIOS**

Projeto de pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Carlos M. S. Figueiredo.

Manaus

2022

Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia - EST

Reitor:

Dr. André Luiz Nunes Zogahib

Vice-Reitor:

Kátia do Nascimento Couceiro

Diretora da Escola Superior de Tecnologia:

Ingrid Sammyne Gadelha Figueiredo

Coordenador do Curso de Engenharia Elétrica:

Israel Gondres Torné

Banca Avaliadora composta por:

Prof. Carlos M. S. Figueiredo, Dr. (Orientador)

Prof. Jozias Parente de Oliveira, Dr.

Prof. Fábio Santos da Silva, Dr.

Data da defesa: 25/10/2022.

CIP – Catalogação na Publicação

Ramos de Sá, Tiago

Embarcações autônomas na região Amazônica: aplicação de técnicas de Aprendizado Profundo para detecção de objetos localizados na superfície de rios / Tiago Ramos de Sá; [orientado por] Prof. Carlos M. S. Figueiredo - Manaus 2022. 65 p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica).
Universidade do Estado do Amazonas, 2022.

1. Detecção de objetos. 2. Embarcações autônomas. 3. Visão Computacional.
- I. Serodio Figueiredo, Carlos Mauricio.

TIAGO RAMOS DE SÁ

**EMBARCAÇÕES AUTÔNOMAS NA REGIÃO AMAZÔNICA:
APLICAÇÃO DE TÉCNICAS DE APRENDIZADO PROFUNDO PARA
DETECÇÃO DE OBJETOS LOCALIZADOS NA SUPERFÍCIE DE RIOS**

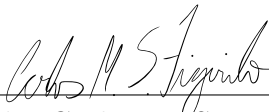
Projeto de pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro Eletricista.

Nota obtida: 10,0 (Dez)

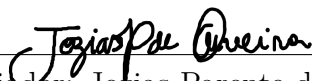
Aprovado em 25/10/2022

Área de concentração: Engenharia Elétrica


BANCA EXAMINADORA



Orientador: Carlos M. S. Figueiredo, Dr.



Avaliador: Jozias Parente de Oliveira, Dr.



Avaliador: Fábio Santos da Silva, Dr.

MANAUS

2022

Dedicatória

Dedico este trabalho a todos os que me ajudaram e contribuíram para meu crescimento acadêmico, profissional e pessoal ao decorrer desta longa caminhada.

AGRADECIMENTO

À Jesus Cristo, que apesar da minha incapacidade de agradá-lo por meios próprios, sempre proveu todas as minhas necessidades para que, dia após dia, eu me mantivesse firme na luta em busca dos meus sonhos.

À minha família, em especial a minha querida mãe, que contribuiu grandemente na construção do meu caráter, da minha educação e do homem que me tornei. Por meio de seus conselhos, aprendi que entregar o meu melhor em tudo que eu fizer não é uma opção, e sim um compromisso.

À minha amada noiva Dayanne Barros Rocha de Mesquita, e em breve esposa, que há anos vem lutando ao meu lado em busca de dias melhores. Ela é diretamente responsável pela minha fonte de motivação, que consiste em me tornar um homem melhor, dia após dia, a fim de fazê-la feliz e sentir que é amada. A ela, minha gratidão por, prontamente, ter me ajudado na criação da base de imagens deste trabalho, contribuindo também grandemente para a finalização desse trabalho. A ela, minha parte mais frágil, minha sincera gratidão. Agradeço também a minha querida sogra, Dayse Barros, por muito apoiar meu crescimento e também me ensinar a tratar sua amada filha como uma verdadeira princesa.

As boas amizades desenvolvidas ao longo da jornada acadêmica no curso de Engenharia Elétrica, em especial aos amigos Eduardo Sales e Isaque Vilson, que sempre estiveram próximos e unidos frentes aos desafios inerentes ao universo acadêmico.

Ao meu orientador, Carlos Mauricio Serodio Figueiredo, pelas orientações técnicas fornecidas e por todas as oportunidades concedidas, frutos deste e de outros trabalhos desenvolvidos.

A todos que participaram, direta ou indiretamente do desenvolvimento deste trabalho de pesquisa, enriquecendo o meu processo de aprendizado. Às pessoas com quem convivi a longo desses anos de curso, que me incentivaram e que certamente tiveram impacto na minha formação acadêmica.

RESUMO

A aplicação de técnicas de Visão Computacional e Aprendizado Profundo para detecção de objetos em superfície de água tem destacado-se como uma forte tendência no contexto de embarcações autônomas. Este trabalho apresenta avaliações de desempenho para detectores de objetos localizados em superfícies marítimas e em superfícies de rios da Amazônia. Para superfícies marítimas, é utilizado a base de imagens *Water Surface Object Detection Dataset* (WSODD), caracterizada por ser uma base publicamente disponível, abrangente e especializada em objetos localizados em superfícies marítimas. Para superfícies de rios da Amazônia, foi criada e anotada a base de imagens *Water Surface Object Detection - Amazon Rivers Dataset* (WSOD-ARD). Para cada um desses conjuntos de imagens, são treinados modelos de detecção de objetos baseados no algoritmo *YOLOv5*. Para o detector treinado com a base de imagens WSODD, este obteve um *Mean Average Precision* (mAP) equivalente à 76.3 %, tendo superado em 11.3 % a precisão média obtida pelo detector CRB-Net nessa mesma base. Para o detector de objetos localizados em superfícies de rios da Amazônia, são utilizadas técnicas de Aprendizado por Transferência a fim de aproveitar os pesos do modelo treinado a partir da base de imagens WSODD, obtendo uma precisão média igual à 75.4 %. Por fim, esse modelo foi integrado em um dispositivo de borda (plataforma embarcada Nvidia Jetson Nano) com o objetivo de simular um cenário real de utilização da aplicação proposta.

Palavras-chave: detecção de objetos, embarcações autônomas, Amazônia, visão computacional, aprendizado profundo, aprendizado de máquina.

ABSTRACT

The application of Computer Vision and Deep Learning techniques in water surface object detection context has been emerged as a strong trend in autonomous vessels scenario. In this work, we present performance comparisons between different object detection models by using two distinct datasets: WSODD (Water Surface Object Detection Dataset) and WSOD-ARD (Water Surface Object Detection - Amazon Rivers Dataset). The first one, WSODD, is characterized by being publicly available, wide and contains objects located on marine water surfaces. The second one, WSOD-ARD, was created and annotated with objects belonging to Amazon rivers water surface scenario. For each one of these datasets, we used YOLOv5 algorithm as architecture for training water surface object detection models. The first model, trained by using WSODD dataset, reached a mAP of 76.3 %, outstanding in 11.3 % the mAP obtained by CRB-Net detector in this benchmark dataset. The second model, trained by using WSOD-ARD dataset, reached a mAP of 75.4 % by using transfer learning techniques. Finally, this model was deployed into an edge device (Nvidia Jetson Nano embedded plataform) in order to simulate a real scenario the proposed application.

Keywords: object detection, self-driving boats, Amazon, computer vision, deep learning, machine learning.

Lista de Figuras

1	Representação de uma imagem em uma matriz de <i>pixels</i>	16
2	Representação genérica de uma máscara.	16
3	Representação da operação de convolução de uma imagem com uma máscara.	17
4	Funcionamento do algoritmo <i>You Only Look Once</i> (YOLO)	19
5	Arquitetura básica do YOLO	19
6	Obtenção da métrica <i>Intersection over Union</i> (IoU)	20
7	Comparação quantitativa e qualitativa entre diferentes valores de IoU.	20
8	Exemplo de imagem coletada para composição da base de imagens. A figura ilustra uma embarcação navegando no rio Solimões.	28
9	Exemplo de imagem coletada para composição da base de imagens. A figura ilustra casas flutuantes, um cenário rotineiro em rios da Amazônia.	29
10	Quadro de imagem pertencente ao vídeo "DE MANAUS A TEFÉ (Parte 2) — Caminhos da Amazônia - AMAZONAS 2022".	31
11	Procedimento utilizado para capturar um quadro de imagem com a ferramenta <i>flameshot</i>	32
12	Visão geral do procedimento utilizado para coletar imagens.	33
13	Processo de anotação de um objeto com a ferramenta Roboflow.	34
14	Exemplo de uma imagem com a presença de oclusão entre objetos.	35
15	Representação de um objeto em larga escala.	36
16	Representação de um objeto em pequena escala.	36
17	Ilustração do Kit de Desenvolvimento <i>Jetson Nano</i>	42
18	Interfaces do Kit de Desenvolvimento <i>Jetson Nano</i>	43
19	Processo de gravação da imagem do sistema operacional Jetson Linux em um cartão de memória com a ferramenta Etcher.	44
20	Periféricos conectados na plataforma <i>Jetson Nano</i> e, ao lado esquerdo, exibição do sistema operacional através de um monitor conectado via cabo HDMI.	45
21	Teste de conexão com o dispositivo embarcado através do comando ping.	46

22	Visão geral dos resultados do treinamento do modelo baseado no detector YOLOv5. Os gráficos (a), (b), (c), (f), (g) e (i) ilustram a curva da função de perda para 3 diferentes variáveis, considerando tanto o conjunto de treino (<i>train</i>), quanto o conjunto de validação. A variação das métricas de precisão (<i>precision</i>) e revocação (<i>recall</i>) são ilustrados pelos gráficos (d) e (e). Por fim, as curvas ilustradas nos gráficos (k) e (l) fornecem resultados de mAP do modelo.	48
23	Matriz de confusão das classificações obtidas pelo modelo.	49
24	Exemplos de objetos detectados após o treinamento do modelo. Na figura (a) nota-se a presença de caixas delimitadoras em volta de um barco e de uma bola (do inglês, <i>ball</i>). Na figura (b), são detectados dois objetos: um barco (<i>boat</i>) e um entulho (<i>rubbish</i>).	50
25	Imagens pertencentes ao <i>dataset WSODD</i> , retratando uma superfície marítima.	52
26	Imagens pertencentes ao <i>dataset</i> regional criado, representando uma superfície típica dos rios da Amazônia.	53
27	Objetos detectados e classificados pelo modelo.	54
28	Visão geral das métricas de performance do modelo treinado a partir da base de imagens regional. Os gráficos (a), (b), (c), (f), (g) e (i) ilustram a curva da função de perda para 3 diferentes variáveis, considerando tanto o conjunto de treino (<i>train</i>), quanto o conjunto de validação. A variação das métricas precisão (<i>precision</i>) e revocação (<i>recall</i>) são ilustrados pelos gráficos (d) e (e). Por fim, as curvas ilustradas nos gráficos (k) e (l) fornecem resultados de mAP do modelo.	55
29	A figura ilustra a variação das métricas de precisão e revocação ao longo da escolha do par (precisão, revocação) desejado	57
30	Matriz de confusão das classificações obtidas pelo modelo.	58
31	Exemplo	59
32	Resultados obtidos a partir da simulação de um cenário real de detecção de objetos com a plataforma embarcada <i>Jetson Nano</i> . À esquerda, a figura ilustra objetos sendo detectados à medida que a câmera, na figura ilustrada à direita, captura novas imagens.	59
33	Ilustração de uma embarcação sendo detectada em um período com menor intensidade de raios luminosos.	60
34	Na figura é ilustrado uma detecção equívoca do modelo, onde ondas formadas pelo deslocamento do barco na superfície da água formaram um objeto com traços semelhantes a de uma canoa.	60

Lista de Tabelas

1	Comparação entre a performance de 18 distintos métodos de detecção de objetos.	22
2	Definição das categorias dos objetos escolhidos para detecção.	30
3	Dados da quantidade de objetos representados por categoria na base de imagens criada.	37
4	Transformações aplicadas nas imagens para o processo de aumento da base imagens de treinamento.	38
5	Comparação entre a performance de 17 distintos métodos de detecção de objetos.	51

Lista de Siglas e Abreviaturas

WSODD *Water Surface Object Detection Dataset*

WSOD-ARD *Water Surface Object Detection - Amazon Rivers Dataset*

YOLOv5 *You Only Look Once Version 5*

mAP *Mean Average Precision*

COLREGs *Convention on the International Regulations for Preventing Collisions at Sea*

SSD *Single-Shot Detector*

Fast-RCNN *Fast Region Based Convolutional Neural Networks*

Mask-RCNN *Mask Region Based Convolutional Neural Networks*

YOLO *You Only Look Once*

CNNs *Convolutional Neural Networks*

IoU *Intersection over Union*

R *Recall*

P *Precision*

AP *Average Precision*

TP *True Positive*

FN *False Negative*

VANTS *Veículos Aéreos Não Tripulados*

MS COCO *Microsoft Common Objects in Context*

SUMÁRIO

INTRODUÇÃO	12
1 REFERENCIAL TEÓRICO	15
1.1 Redes Neurais Convolucionais	15
1.1.1 A operação de convolução	15
1.1.2 Mapa de Características	17
1.2 Técnicas utilizadas em detecção de objetos	18
1.2.1 YOLO	18
1.3 Métricas de avaliação utilizadas para detecção de objetos	19
1.3.1 Interseção sobre União (IoU)	19
1.3.2 Revocação, Precisão e Precisão Média (<i>Average Precision</i> (AP))	20
1.3.3 Média da Precisão Média (mAP)	21
2 TRABALHOS RELACIONADOS	21
3 MATERIAIS E MÉTODOS	24
3.1 Caracterização da pesquisa aplicada	25
3.2 Modelo para detecção de objetos a partir da base de imagens <i>WSODD</i>	26
3.2.1 Base de imagens <i>WSODD</i>	26
3.2.2 Validação do modelo	27
3.3 Modelo para detecção de objetos localizados na superfície de rios da Amazônia (<i>dataset WSOD-ARD</i>)	27
3.3.1 Aquisição de imagens	27
3.3.2 Anotação de objetos	33
3.3.3 Técnicas de pré-processamento de imagens	37
3.3.4 Validação do modelo	38
3.3.5 Treinamento do detector de objetos	39
3.3.6 Detalhes de implementação	40

3.3.7	Integração do modelo em um dispositivo embarcado	41
4	RESULTADOS	47
4.1	Avaliação de performance do modelo treinado a partir da base WSODD . . .	47
4.1.1	Análise da função de perda	47
4.1.2	Análise das métricas precisão e revocação	48
4.2	Comparações entre performances de modelos similares	50
4.3	Avaliação de performance do modelo treinado a partir da base de imagens WSOD-ARD	54
4.3.1	Análise da função de perda	54
4.3.2	Análise da precisão e revocação	56
4.3.3	Análise de performance do modelo em uma plataforma embarcada	58
5	CONCLUSÃO	61
	REFERÊNCIAS	62

INTRODUÇÃO

Embarcações que operam sem a intervenção humana são conhecidas como embarcações autônomas. Em conformidade com as regulamentações da *Convention on the International Regulations for Preventing Collisions at Sea* (COLREGs)¹, evitar colisões com objetos estáticos ou dinâmicos, sem que haja intervenção humana, de maneira geral, é um dos requisitos mais importantes para a viabilização de embarcações autônomas (GU et al., 2019). Assim, no contexto de rios navegáveis da Amazônia, elas podem ser desenvolvidas como uma infraestrutura flexível, isto é, uma ponte temporária ou uma estação flutuante, evitando obstáculos à medida que navega pela superfície da água de forma autônoma (WANG et al., 2020). Além disso, podem também ser utilizadas em contextos onde há uma demanda por vigilância militar (PRASAD et al., 2017), controle de tráfego marítimo (PRASAD et al., 2020) e proteção ambiental (BLOISI et al., 2012, 2011). Esses exemplos representam algumas das muitas aplicações possíveis à medida que métodos de Visão Computacional e técnicas de Inteligência Artificial, com vistas à detecção de objetos localizados em superfície de rios da Amazônia, são desenvolvidos (XU et al., 2017).

Existem muitos algoritmos especializados em detecção de objetos, tais como *Single-Shot Detector (SSD)* (LIU et al., 2015), *RetinaNet* (LIN et al., 2017), *Fast Region Based Convolutional Neural Networks (Fast-RCNN)* (GIRSHICK, 2015), *Mask Region Based Convolutional Neural Networks (Mask-RCNN)* (HE et al., 2017), *YOLO* (LI et al., 2020) e entre outros. Especialmente em relação ao algoritmo *YOLO*, sua quinta versão, *You Only Look Once Version 5 (YOLOv5)*, encontra-se em estado da arte e tem se destacado pela sua praticidade e performance frente a outros detectores clássicos. Além disso, foi pouco explorado para contextos de objetos localizados em superfície de água. Por último, *YOLOv5* incorpora técnicas de aumento de imagens que são aplicadas em tempo real de treinamento, provendo modelos robustos e que generalizam melhor. Em virtude desses motivos, foi escolhido como base para os modelos de detecção de objetos avaliados neste trabalho.

Ainda que haja promissores detectores de objetos em estado da arte disponíveis conforme mencionado acima, há uma significativa escassez de abordagens baseadas em ar-

¹Para mais detalhes, acessar: (<https://www.imo.org/en/OurWork/Safety/Pages/Preventing-Collisions.aspx>)

quitaturas convolucionais que são especializadas na detecção de objetos localizados em superfície de água (PRASAD et al., 2017). Isso ocorre, principalmente, em virtude de existirem poucas bases de imagens disponíveis publicamente que retratam objetos em superfície de água (ZHOU et al., 2021). Das poucas bases disponíveis, a maioria é caracterizada por não ser representativa em relação à quantidade de instâncias por categoria. Além disso, muitas delas apresentam pouca variedade no que diz respeito às condições de captura das imagens. *ImageNet* (DENG et al., 2009), *Microsoft Common Objects in Context (MS COCO)* (LIN et al., 2014) e *Places 2* (ZHOU et al., 2018) são exemplos de algumas bases de imagens públicas que representam os problemas citados anteriormente, pois, apesar de possuírem grande volume de imagens, são bastantes escassas no que tange a objetos em superfície de água. Esse cenário representa um notável problema, pois impacta negativamente para o fomento ao desenvolvimento de técnicas de detecção de objetos em superfície de água.

Para testar, comparar e otimizar os métodos de detecção para o contexto de objetos localizados em superfície de água, é fundamental que volumosas bases de imagens anotadas estejam disponíveis publicamente. Caso essa condição não seja tangível, é possível optar pela abordagem da criação de uma base de imagens própria, sendo essa alternativa, a priori, muitas vezes inviável a curto ou médio prazo. De todo modo, tendo sido obtida uma base de referência, diferentes abordagens de detecção de objetos são testados, permitindo a comparação e otimização de performance em métricas de acurácia e tempo de inferência.

Próximo ao período em que este trabalho foi desenvolvido, foi disponibilizada uma base de imagens anotadas que retratam, com notável qualidade e variedade, objetos localizados em superfície de água. Essa base, intitulada como *WSODD (Water Surface Object Detection Dataset)* (ZHOU et al., 2021), mitiga os pontos levantados anteriormente e será utilizada como referência para comparar a performance do detector *YOLOv5* com diferentes detectores clássicos de objetos. Em suma, a base *WSODD* é caracterizada por possuir uma quantidade significativa de instâncias por categorias que são comuns no contexto de objetos localizados em superfície da água. É composta por 7.467 imagens, diversificadas em 3 ambientes físicos distintos, isto é, oceanos, lagos e rios; 3 condições de iluminação distintas - ao amanhecer, ao entardecer e ao anoitecer; 3 condições climáticas, isto é, dias ensolarados, nublados e nebulosos. Ao todo, existem 14 classes e 21.911 objetos anotados.

Considerando o que foi exposto acima, as principais contribuições deste artigo são:

(I) Avaliar a performance da versão mais recente do detector *YOLO (YOLOv5)* na base *WSODD*, considerando as principais métricas de avaliação utilizadas em modelos de detecção de objetos. Ao final, verificou-se que o modelo treinado para detectar objetos em superfície de água alcançou ótima capacidade de generalização frente a diferentes categorias.

(II) Comparar a performance do detector *YOLOv5* com outros diferentes métodos

de detecção apresentados no trabalho de (ZHOU et al., 2021), utilizando a mesma base de imagens *WSODD*. Ao final, observou-se que o método *YOLOv5* superou significativamente todas as outras abordagens comparadas.

(III) Treinar e avaliar um modelo baseado no algoritmo *YOLOv5* a partir da base de imagens *WSOD-ARD* (*Water Surface Object Detection - Amazon Rivers Dataset*) - uma base regional criada ao longo do desenvolvimento deste trabalho, sendo caracterizada por representar cenários de objetos localizados em superfícies de rios da Amazônia.

(IV) Avaliar a performance de inferência dos modelos de detecção de objetos obtidos em uma plataforma embarcada. Ao final, foi possível realizar a avaliação de performance do modelo em um dispositivo de borda, isto é, um dispositivo embarcado que possui autonomia de processamento para aplicações de Visão Computacional e Inteligência Artificial.

Sob a ótica da proposta deste trabalho, Seção 1 apresenta brevemente aspectos essenciais de fundamentação teórica que norteiam as bases deste trabalho. Em seguida, Seção 2 descreve sobre os trabalhos relacionados e como este trabalho destaca-se em relação aos demais. Seção 3 contempla os métodos empregados para alcançar os resultados obtidos. Seção 4 apresenta uma discussão sobre os resultados alcançados. Por fim, Seção 5 apresenta considerações finais sobre o que foi desenvolvido.

1 REFERENCIAL TEÓRICO

Esta seção apresenta uma sumarização dos principais fundamentos teóricos relacionados à proposta do trabalho. A Seção 1.1 apresenta uma visão geral sobre conceitos elementares em Redes Neurais Convolucionais como base para as técnicas de detecção de objetos em imagens. Em seguida, na Seção 1.2, são apresentadas as principais técnicas e métodos em estado da arte adotados para algoritmos de detecção de objetos, em especial, para detecção de objetos em superfície d'água. A Seção 1.3, por fim, lista e explana sobre as principais métricas de avaliação utilizadas no tema proposto.

1.1 Redes Neurais Convolucionais

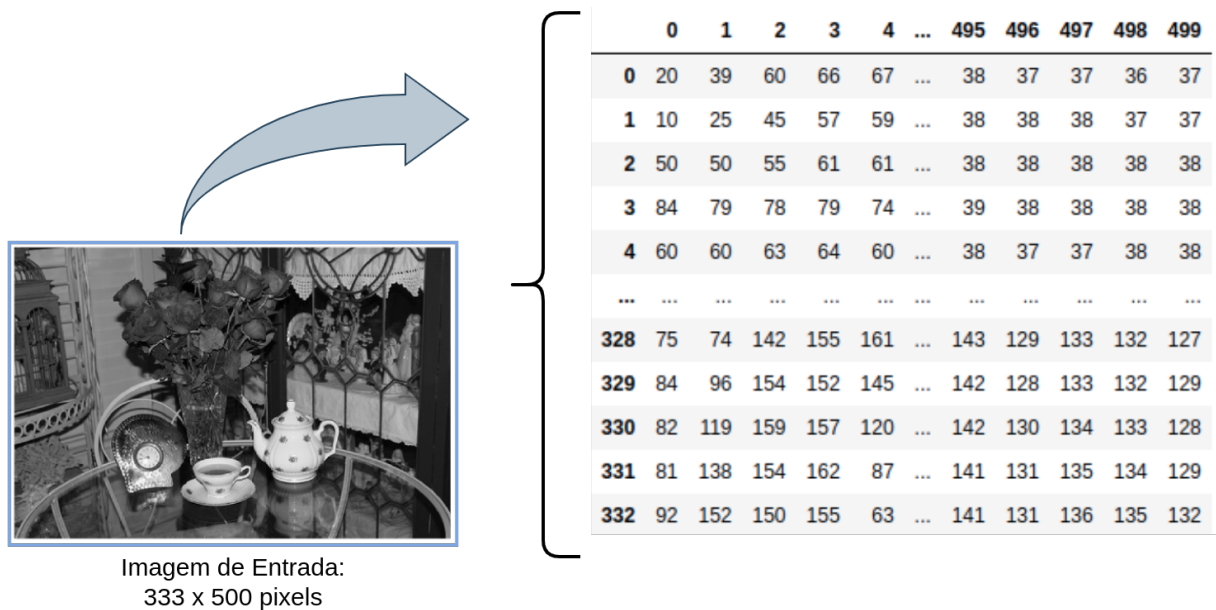
Redes Neurais Convolucionais (*Convolutional Neural Networks* (CNNs)) (LECUN et al., 2001) são arquiteturas de Aprendizado Profundo comuns em algoritmos de Visão Computacional, especialmente em detectores de objetos. Em termos gerais, CNNs são redes neurais simples que utilizam, em pelo menos uma de suas camadas, uma operação conhecida por convolução (GOODFELLOW; BENGIO; COURVILLE, 2016). Além disso, CNNs possuem camadas que são imprescindíveis para o processamento de tarefas cujos dados de entrada são imagens, tais como, por exemplo, filtros detectores de características, mapas de características, e entre outras. Por meio dessas camadas, é possível implementar algoritmos robustos para tarefas preditivas inseridas no contexto de Visão Computacional, em especial à detecção de objetos.

1.1.1 A operação de convolução

A operação de convolução, no campo da Visão Computacional, representa, de forma geral, realizar a convolução de uma máscara com uma imagem.

Para exemplificar, seja uma imagem I de tamanho 5×5 definida como uma matriz de pixel da seguinte forma:

Figura 1 – Representação de uma imagem em uma matriz de *pixels*.



Fonte: (Elaborado pelo autor, 2022).

Para que seja possível extrair características dessa imagem, é definido um *kernel* (filtro) K cujo tamanho é 3×3 :

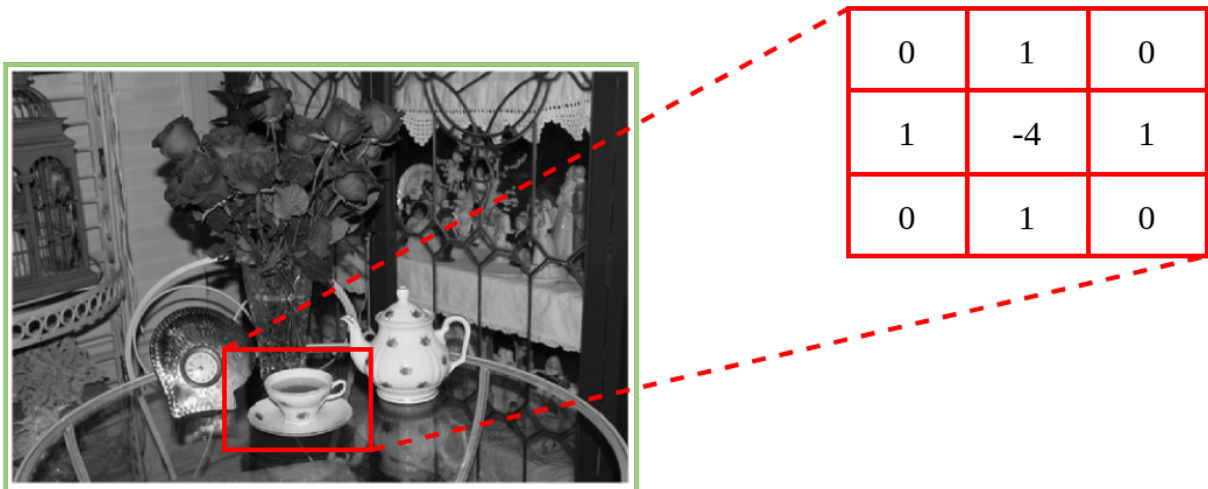
Figura 2 – Representação genérica de uma máscara.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Fonte: (Elaborado pelo autor, 2022).

Essas características podem ser interpretadas como padrões visuais que ajudam ou não a identificar ou classificar um determinado objeto. Por exemplo, o *kernel* K_1 ilustrado na Figura 3 representa um filtro detector de bordas. Ao realizar a operação de convolução da imagem representada por uma xícara com o *kernel* K_1 , é possível obter as bordas desse objeto.

Figura 3 – Representação da operação de convolução de uma imagem com uma máscara.



Fonte: (Elaborado pelo autor, 2022).

Em síntese, a operação de convolução de filtros com imagens é definida matematicamente conforme a equação a seguir. Seja I uma matriz de *pixels* representando uma imagem e K um *kernel* representando um filtro qualquer, a resposta da convolução S é:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n)$$

O resultado da operação de convolução para filtros com diferentes pesos e tamanhos viabiliza a construção do mapa de características, conforme será visto a seguir.

1.1.2 Mapa de Características

Em arquiteturas baseadas em Redes Neurais Convolucionais, a resposta do processamento realizado em um determinado conjunto de entrada multidimensional, por meio da operação de convolução, são os mapas de características. Dessa forma, ao processar uma imagem com milhões de *pixels*, apenas os contornos horizontais ou verticais que são significativos para o contexto da aplicação são armazenados em mapas de características, reduzindo o processamento de milhões de *pixels* em centenas de *pixels*.

Existem muitas vantagens nessa abordagem, principalmente no que diz respeito à redução do processamento. Por meio de arquiteturas baseadas em CNNs, as interações entre os neurônios são esparsas, viabilizando uma relevante redução de uso de espaço em memória para os modelos, além de melhorar a eficiência estatística.

Redes Neurais Convolucionais possuem características que as tornam efetivas comparadas com outras arquiteturas, a saber, interação esparsa, compartilhamento de parâmetros e representações equivariantes.

1.2 Técnicas utilizadas em detecção de objetos

Detecção de objetos é um ramo da Visão Computacional que é responsável por inferir a localização de um objeto em uma dada imagem, junto com a sua respectiva classe (ZHAO et al., 2018). De maneira geral, os métodos mais comuns que auxiliam nessa tarefa podem ser categorizados em dois principais tipos: métodos de passada única e métodos de passada dupla.

Métodos de passa única priorizam o tempo de resposta da inferência. Os detectores YOLO (LI et al., 2020), SSD (LIU et al., 2015) e RetinaNet (LIN et al., 2017) são exemplos desse método. Por outro lado, métodos de passada dupla priorizam a acurácia da detecção. Alguns algoritmos destacam-se para esse método, tais como *Fast R-CNN* (GIRSHICK, 2015), *Faster R-CNN* (REN et al., 2015), *Mask-RCNN* (HE et al., 2017) e entre outros.

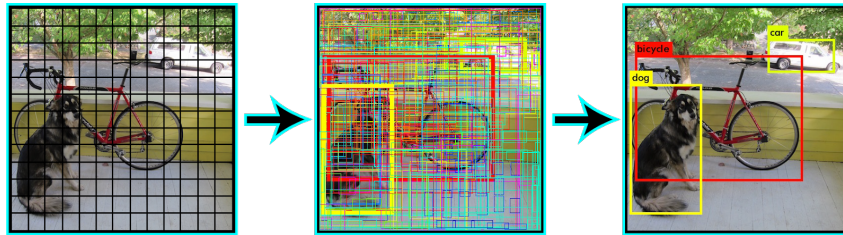
1.2.1 YOLO

YOLO é um método baseado em Redes Neurais Convolucionais que realiza detecção de objetos com uma passada única. Ele foi idealizado para sistemas que necessitam realizar inferências em um intervalo de tempo curto, enquanto mantém uma acurácia alta. Para isso, ele aborda a detecção de objetos como um único problema de regressão, onde, diretamente a partir dos *pixels* de uma dada imagem, extrai as coordenadas das caixas delimitadoras, juntamente com as probabilidades das classes.

O funcionamento básico do algoritmo é dividir uma imagem de entrada em N grades, cada uma com uma região dimensional igual à $S \times S$. Assim, para cada objeto presente na imagem, uma célula da grade é "responsável" pela detecção e localização do objeto. Por exemplo, na Figura 4, várias predições são obtidas para o mesmo objeto, com diferentes caixas delimitadoras. Contudo, YOLO utiliza a técnica Supressão de Não Máximos para reduzir a quantidade de caixas delimitadoras, mantendo apenas aquelas com pontuação de confiança acima do limiar definido. Esse processo se repete até encontrar a caixa delimitadora com a maior probabilidade de corretamente localizar e classificar um determinado objeto.

YOLO utiliza a técnica Supressão de Não Máximos para reduzir a quantidade de caixas delimitadoras, mantendo apenas aquelas com pontuação de confiança acima do limiar definido. Esse processo se repete até encontrar a caixa delimitadora com a maior probabilidade de corretamente localizar e classificar um determinado objeto.

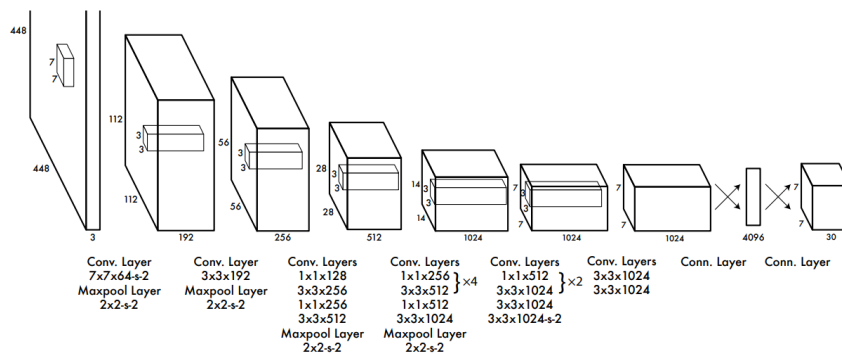
Figura 4 – Funcionamento do algoritmo YOLO



Fonte: (LI et al., 2020).

A sua arquitetura é inspirada na arquitetura *GoogLeNet*, e possui um total de 24 camadas convolucionais com 2 camadas totalmente conectadas ao final. As camadas convolucionais utilizam Leaky ReLU como função de ativação para inserir comportamentos não-lineares na rede. A Figura 5 ilustra sua arquitetura, que é baseada em redes neurais convolucionais.

Figura 5 – Arquitetura básica do YOLO



Fonte: (LI et al., 2020).

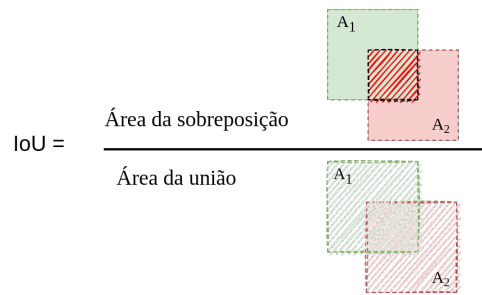
1.3 Métricas de avaliação utilizadas para detecção de objetos

As métricas de avaliação de modelos detectores de objetos que serão utilizadas ao decorrer deste trabalho são: Interseção sobre União (IoU), Revocação (*Recall* (R)), Precisão (*Precision* (P)), Precisão Média (AP) e Média da Precisão Média (mAP). Essas métricas possuem outras variações, dependendo da base de imagens para a qual foram utilizadas em competições passadas.

1.3.1 Interseção sobre União (IoU)

Interseção sobre União é conhecida como uma métrica relevante para medir a sobreposição entre duas caixas delimitadoras. A Figura 6 ilustra como é obtido essa métrica.

Figura 6 – Obtenção da métrica IoU



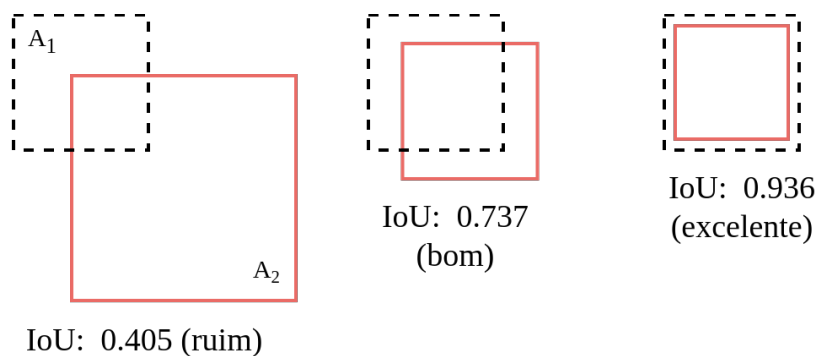
Fonte: (Elaborado pelo autor, 2022).

Quanto mais próximo IoU de 1, maior a probabilidade do objeto detectado ser um verdadeiro positivo; do contrário, quanto mais próximo de 0, maior a probabilidade do objeto detectado ser um verdadeiro negativo (ou falso positivo). A Figura 7 ilustra possíveis valores quantitativos e qualitativos (ruim, bom e excelente).

Dado a ilustração acima, um objeto detectado é considerado verdadeiro positivo (*True Positive* (TP)) dependendo do limiar IoU escolhido. Por exemplo, se o IoU é maior que 0.5, o objeto é classificado como TP para o *dataset Pascal VOC*; se for menor, é classificado como falso positivo (FP).

Por outro lado, o desafio COCO avalia os objetos classificados sob vários limiares de IoU na faixa de 0.05 até 0.95.

Figura 7 – Comparação quantitativa e qualitativa entre diferentes valores de IoU.



Fonte: (Elaborado pelo autor, 2022).

1.3.2 Revocação, Precisão e Precisão Média (AP)

Precisão mede a porcentagem de verdadeiros positivos sobre o total de objetos detectados (verdadeiros positivos e falsos positivos). É definido matematicamente como:

$$\text{Precisão} = \frac{TP}{TP + FP} = \frac{\text{objectos detectados corretamente}}{\text{todos os objetos detectados}} \quad (1)$$

A Revocação, por outro lado, mede a porcentagem de verdadeiros positivos sobre o total de casos relevantes dentro de um conjunto de dados. É definido matematicamente como:

$$\text{Revocação} = \frac{\text{TP}}{\text{TP} + \text{False Negative}(\text{FN})} = \frac{\text{objectos detectados corretamente}}{\text{todos os objetos do ground truth}} \quad (2)$$

Por meio da precisão média, busca-se medir as detecções corretas sob diferentes limites do IoU. Para isso, são coletadas as predições realizadas pelo modelo e elencando elas em um intervalo de confiança muito alta até um intervalo de confiança muito baixa. Após essa etapa, é calculado o valor médio da precisão para diferentes valores de Revocação (R). A Precisão Média pode ser obtida a partir da seguinte equação matemática:

$$\text{AP} = \frac{1}{11} \sum_{\text{recall}=i}^h \max(\text{precision}(\text{recall}_i)) \quad (3)$$

1.3.3 Média da Precisão Média (mAP)

A mAP é a média da AP entre todas as classes. Para o *dataset* COCO, a mAP é calculada da seguinte forma:

$$\text{mAP}_{\text{COCO}} = \frac{\text{mAP}_{0.50} + \text{mAP}_{0.55} + \dots + \text{mAP}_{0.95}}{10} \quad (4)$$

em que $\text{AP}@[.5 : .05 : .95]$ representa os limites de IoU com stepsize de .05

Existem ainda outras variações do mAP como, por exemplo, cálculo da mAP para diferentes tamanhos das áreas ocupadas pelos objetos.

2 TRABALHOS RELACIONADOS

Detecção de objetos em superfície de água, especialmente em navegações marítimas, compreende um cenário de notável desafio em virtude da escassez mesclada com a complexidade das imagens observadas. A seguir, são apresentados alguns trabalhos de destaque relacionados ao tema de detecção de objetos no contexto de embarcações autônomas.

A base de imagens mencionada na Seção Introdução foi introduzida pelo trabalho dos autores em (ZHOU et al., 2021). Nesse trabalho, destaca-se a base de imagens *WSODD* como uma contribuição muito significativa para o cenário de detecção de objetos em superfície de água. Ela é caracterizada por possuir uma quantidade significativa de instâncias por categorias, sendo composta por 7.467 imagens, diversificadas em 3 ecossistemas distintos, isto é, oceanos, lagos e rios; 3 condições de iluminação distintas - ao

amanhecer, ao entardecer e ao anoitecer; 3 condições climáticas, isto é, dias ensolarados, nublados e nebulosos. Ao todo, existem 14 classes e 21.911 objetos anotados. Além disso, destacam-se também os testes comparativos que foram realizados com diferentes métodos de detecção de objetos. Ao todo, 17 detectores foram testados e avaliados em função da métrica *mAP* (*Mean Average Precision*). Tanto a base de imagens quanto as métricas obtidas foram utilizadas como referência para comparar com os resultados obtidos neste presente trabalho.

Tabela 1 – Comparação entre a performance de 18 distintos métodos de detecção de objetos.

Rótulo	Imagens	Instâncias
<i>Animal</i>	50	94
<i>Tree</i>	72	219
<i>Grass</i>	103	110
<i>Buoy</i>	153	167
<i>Mast</i>	177	354
<i>Person</i>	357	695
<i>Rubbish</i>	461	669
<i>Platform</i>	480	614
<i>Ball</i>	652	2,609
<i>Rock</i>	696	154
<i>Harbor</i>	1,211	1,224
<i>Bridge</i>	1,827	2,014
<i>Ship</i>	1,832	3,423
<i>Boat</i>	4,325	8,179
Total	7,467	21,911

Fonte: (ZHOU et al., 2021)

No trabalho dos autores em (BLOISI et al., 2012), é proposto um *framework* para detecção de embarcações em ambientes marítimos com base em um *dataset* (em Português, base de dados) composto por imagens capturadas por câmeras. Ele torna possível o aprimoramento das funcionalidades dos atuais sistemas de Serviços de Tráfego de Embarcações (VTS), alcançando também regiões habitadas, onde sistemas baseados em radares não poderiam ser utilizados em virtude das emissões e radiações eletromagnéticas. O detector utilizado é baseado no método *Haar-like*, conhecido, originalmente, para finalidades de reconhecimento de faces. Os resultados do experimento foram realizados em cima de um conjunto de teste composta por apenas 163 imagens. Por fim, ressalta-se que o trabalho contempla apenas objetos pertencentes a classe de embarcações.

Para detecção de objetos localizados na superfície do mar, os autores em (BLOISI et al., 2012) propõem um *framework* que utiliza uma base de dados criada a partir de imagens capturadas por câmeras fixas em embarcações. O detector utilizado é baseado

no método *Haar-like*², e tem como principal finalidade detectar embarcações. Os testes com esse detector foram realizados em uma base composta por apenas 163 imagens que retratam somente objetos pertencentes à categoria embarcação.

No trabalho dos autores em (PRASAD et al., 2017), destaca-se a exploração de diferentes abordagens para a detecção de objetos, considerando o contexto de embarcações autônomas. No geral, são apresentados detectores do tipo duas etapas que contemplam três etapas, a saber, (i) detecção de horizontes (baseado em projeção, baseado em região e abordagem híbrida), (ii) subtração de fundo estático e (iii) segmentação *foreground*. As análises de performance desses detectores são realizadas por meio do *Singapore Maritime Dataset*³. Por fim, observa-se que os trabalhos apresentados exploram apenas ambientes marítimos, não abrangendo outras superfícies de água, tais como rios, canais, lagoas e entre outros.

Em (BLOISI et al., 2012), é proposto a base de imagens *MarDTC*, sendo esta criada a partir do sistema *ARGOS* (*Automatic Remote Grand Canal Observation System*) e de bases públicas, cuja principal finalidade é viabilizar a avaliação de performance de técnicas de Visão Computacional orientadas a detecção, classificação e rastreamento de embarcações. Ressalta-se que as imagens são majoritariamente compostas por diferentes tipos de embarcações navegando em um trecho localizado no Grande Canal de Veneza. Ao decorrer do trabalho, é proposto um classificador baseado em técnicas de Aprendizado de Máquina para categorizar diferentes classes de embarcações.

Por último, os autores em (RIBEIRO et al., 2019) apresentam uma base de imagens anotadas, capturadas a partir de Veículos Aéreos Não Tripulados (VANTS) A principal finalidade é de possibilitar vigilância em um contexto marítimo. Para isso, o trabalho propõe alguns experimentos voltados à detecção de embarcações. Para isso, utiliza detectores clássicos do tipo passada dupla, em especial os métodos *R-CNN* e *Blob*, além de abordagens não supervisionadas. De maneira geral, nota-se que as imagens capturadas são restritas a ambientes marítimos, especializadas apenas para embarcações e em escalas muito pequenas.

Por fim, em um outro trabalho relacionado ao contexto de detecção de objetos em superfícies de água, os autores (RIBEIRO et al., 2019) apresentam um *dataset* de imagens anotadas, capturadas a partir de drones (veículos aéreos não tripulados) com a finalidade de viabilizar vigilâncias em ambientes marítimos. As imagens possuem instâncias de objetos pertencentes a diferentes tipos de embarcações, tais como cargueiro, barcos de passeio e balsas salva-vidas. Em seguida, o trabalho propõe alguns experimentos com essa base de imagens, em especial, detecção de embarcações. Para isso, utiliza detectores

²Mais informações sobre esse algoritmo, acessar: https://en.wikipedia.org/wiki/Haar-like_feature

³Mais detalhes sobre essa base de imagens, acessar: <https://sites.google.com/site/dilipprasad/home/singapore-maritime-dataset>

clássicos do tipo duas etapas, similares aos métodos *-CNN* e *Blob*, e abordagens não supervisionadas. Por fim, nota-se que as imagens capturadas são restritas a ambientes marítimos, especializadas apenas para embarcações e em escalas muito pequenas.

Com exceção do primeiro trabalho mencionado, o restante compartilha uma característica em comum, isto é, as bases de imagens utilizadas não são representativas de objetos localizados em superfície de água. Em síntese, as imagens retratam, majoritariamente, apenas embarcações, não abrangendo, todavia, outros possíveis tipos de objetos, tais como rochas, vegetação, portos, entulhos, animais, plataformas de embarque, boias, pontes e entre outros. Além disso, as imagens retratam apenas embarcações em cenários marítimos, não contemplando outros ecossistemas aquáticos, isto é, lagos, lagoas, rios, geleiras e entre outros. Por fim, este trabalho destaca-se dos demais por avaliar e comparar, em primeira mão, a performance de um detector em estado de arte, *YOLOv5*, em uma base de imagens com notável representatividade em diferentes objetos e diferentes ecossistemas aquáticos.

3 MATERIAIS E MÉTODOS

Esta seção detalha sobre os materiais e procedimentos que foram utilizados a fim de desenvolver um detector de objetos localizados em superfícies de rios da Amazônia para o contexto de embarcações autônomas. Ela é dividida em cinco principais partes, a saber: (i) caracterização da pesquisa aplicada, (ii) modelo para detecção de objetos a partir da base de imagens *WSODD*, (iii) modelo para detecção de objetos localizados na superfície de rios da Amazônia (base *WSOD-ARD*), (iv) detalhes de implementação e (v) integração entre o modelo treinado e a plataforma embarcada *Jetson Nano*

No que diz respeito ao item (i), em se tratando de uma pesquisa aplicada, este trabalho foi desenvolvido a partir de pesquisa exploratória, seguindo uma abordagem hipotético-dedutivo e o método de procedimento monográfico em sua elaboração. Para fundamentar a pesquisas realizada, foi utilizado o procedimento técnico de pesquisa bibliográfica, bem como o uso de documentação indireta, com auxílio de documentos primários e secundários. Ao final, os resultados da pesquisa aplicada foram discutidos e interpretados de forma quantitativa.

Antes de desenvolver um modelo específico para detectar objetos localizados na superfície dos rios da Amazônia, uma primeira etapa consistiu-se em implementar e avaliar um detector de objetos localizados em superfície de água em distintos ecossistemas. Essa etapa será tratada a partir do item (ii) e será vista com maior riqueza de detalhes posteriormente.

Em relação ao item (iii), detecção de objetos localizados na superfície de rios da Amazônia, para essa etapa, foram utilizados métodos de criação da base de imagens,

bem como do implementação do modelo e avaliação do mesmo, considerando diferentes métricas de performance.

Para a implementação dos modelos, foram utilizados ambientes de desenvolvimentos configurados em contêineres. Além disso, foram definidos hiperparâmetros para o treinamento de cada um dos modelos. Esses detalhes serão detalhados adiante.

Por último, a integração entre o modelo treinado e a plataforma embarcada *Jetson Nano* - item (v), permitiu uma simples simulação onde um dispositivo de borda é utilizado para realizar, em tempo real de execução, a detecção de objetos localizados em superfícies de rios da Amazônia por meio de imagens e vídeos reais capturados a partir de fontes externas. Logo em seguida, serão apresentados os pormenores oriundos do desenvolvimento de cada uma das etapas mencionadas acima.

3.1 Caracterização da pesquisa aplicada

Foram feitas consultas em materiais bibliográficos diversos para o levantamento de técnicas em estado da arte, relacionadas ao tema proposto pelo trabalho. Assim, foram realizadas pesquisas sobre o campo da Inteligência Artificial, especialmente sobre Redes Neurais Profundas, métodos de detecção de objetos, técnicas de Processamento Digital de Imagens e embarcações autônomas. Os resultados dessas pesquisas permitiram obter parâmetros de referência para comparação e verificação dos resultados que foram obtidos ao decorrer deste trabalho.

Ao decorrer do processo de pesquisa, foram estabelecidos os seguintes objetivos: (i) identificar referenciais teóricos que relacionam modelos de detecção de objeto localizados em superfícies de diferentes ecossistemas (rios, mares, oceanos, lagoas e entre outros) com técnicas baseadas em algoritmos de aprendizado profundo e algoritmos de visão computacional; (ii) identificar algoritmos de detecção de objetos em estado da arte para o cenário proposto de objetos localizados em superfícies aquáticas; (iii) encontrar bases de imagens publicamente disponíveis na *Internet* que retratassem o contexto de navegação fluvial e embarcações. Esses três principais objetivos nortearam os passos da pesquisa aplicada.

Dentre as fontes utilizadas para a busca, destacam-se as seguintes bibliotecas digitais: *ACM Digital Library*, *IEEE Software*, *SCOPUS*, *Google Scholar* e *Google Search*. Em cada uma dessas fontes, foi estabelecido um procedimento simples de revisão de literatura com as seguintes abordagens:

- População: "autonomous vehicles"OU "self-driving boats"OU "self-driving vehicles"OU "self-driving vessels"OU "autonomous boats"OU "autonomous vessels", "barcos autônomos"OU "embarcações autônomas"OU "boat"OU "vessel";

- Intervenção: "water surface object detection"OU "deteccção de objetos em superfície de água"OU "deteccção de objetos em superfícies"
- Resultado: "deep learning"OU "visão computacional"OU "computer vision"OU "object detection"OU "aprendizado profundo"OU "machine learning"OU "aprendizado de máquina"

Conforme listado acima, o termo população refere-se ao conjunto de palavras que descrevem, de forma bem ampla, o tema de interesse. Em seguida, por meio da intervenção, é feito uma delimitação do tema, contemplando palavras-chaves que direcionam o tema para um subtema específico de interesse. Por último, em resultado, são utilizadas palavras-chaves que filtram o tema delimitado para trabalhos que utilizam técnicas de visão computacional ou inteligência artificial.

Por meio das palavras-chaves listadas acima, foram encontrados referenciais teóricos relevantes, dentre os quais alguns foram apresentados na Seção 2. Por fim, os materiais encontrados deram base para o levantamento de técnicas em estado da arte relacionadas ao tema proposto pelo trabalho, contribuindo também para a comparação entre resultados de diferentes detectores de objetos localizados em superfície de água. Dentre os achados, destacam-se os resultados obtidos na Seção 3.2, onde a base de imagens é reaproveitada para o desenvolvimento de uma parte deste presente trabalho.

3.2 Modelo para deteção de objetos a partir da base de imagens *WSODD*

Com vistas ao objetivo de avaliar e comparar a performance do algoritmo *YOLOv5* para deteção de objetos localizados em superfície de água, foram considerados três principais etapas para composição do experimento, isto é, (i) formatação das anotações da base de imagens, (ii) definição da abordagem para validação do modelo e (iii) especificação do ambiente de desenvolvimento. A seguir, são descritas em detalhes cada uma dessas etapas.

3.2.1 Base de imagens *WSODD*

A base de imagens *WSODD* foi criada com o propósito de contribuir para o contexto de deteção de objetos localizados em superfície de diferentes ecossistemas aquáticos. É uma base com imagens anotadas, fornecendo informações tais como coordenadas da localização e categoria objeto. Todas essas anotações são dados imprescindíveis para o treinamento de modelos orientados à deteção de objetos. Ainda, é composto por uma considerável quantidade de imagens distribuídas entre diferentes ambientes, diferentes estações do dia e condições climáticas diversas. Enfim, até o momento da publicação deste trabalho, essa havia sido a base disponibilizada publicamente que possuía a maior abrangência em função das características acima mencionadas.

As anotações das imagens pertencentes à base *WSODD* foram disponibilizadas em formato *PASCAL VOC* (EVERINGHAM et al., 2015), não sendo possível, portanto, utilizá-las para o treinamento de modelos a partir do método *YOLO*, pois este requer que as anotações estejam em formato *YOLO*. Dessa forma, foi realizado a implementação de um algoritmo, escrito em *Python*, apto a converter as anotações do formato de origem, *PASCAL VOC*, para o formato de destino, *YOLO*.

3.2.2 Validação do modelo

No que tange a validação do modelo, foi utilizado o método *holdout* para a validação cruzada do experimento. Dessa forma, foi mantido a mesma proporção utilizada em (ZHOU et al., 2021), ou seja, 70 % das imagens foram destinadas para o conjunto de treinamento, enquanto que o restante, 30 %, foram utilizadas no conjunto de teste. Ressalta-se que a seleção delas, em ambos os conjuntos, foi realizada de forma aleatória, mantendo uma distribuição representativa das classes em cada conjunto.

3.3 Modelo para detecção de objetos localizados na superfície de rios da Amazônia (*dataset WSOD-ARD*)

A implementação do modelo para detecção de objetos localizados na superfície de rios da Amazônia foi dividida em 5 etapas, isto é: (i) aquisição de imagens, (ii) anotação de objetos, (iii) técnicas de pré-processamento de imagens, (iv) definição da estratégia validação do modelo e (v) treinamento do detector de objetos.

3.3.1 Aquisição de imagens

A primeira etapa foi destinada a criação de uma base de imagens que retratam objetos comumente localizados na superfície de rios da Amazônia, sob distintas condições de captura. Conforme será visto a seguir, essa etapa apresentou alguns desafios, em especial no que diz respeito às estratégias utilizadas para coletar imagens de interesse.

De forma objetiva, não foi possível capturar imagens diretamente na região de interesse da pesquisa, isto é, imagens capturadas a partir de câmeras de alta resolução, posicionadas nas embarcações de modo que pudessem retratar um cenário real das superfícies de rios da Amazônia. A impossibilidade dessa opção, em síntese, ocorreu em virtude da escassez recursos, tanto de mão de obra, quanto financeiro. No mínimo, seria necessário uma embarcação própria para que fosse possível implantar uma infraestrutura capaz de capturar, armazenar e disponibilizar as imagens. Além disso, seria necessário a cooperação de voluntários para que fosse possível planejar um revezamento a fim de capturar imagens que retratassem a região de interesse em distintas condições de captura, isto é, em diferentes intensidades de iluminação ao longo de um dia (por exemplo, período noturno, matutino e vespertino), em diferentes condições climáticas (por exemplo,

períodos climáticos caracterizados pela presença de neblina, chuva, sol e entre outros) e em distintas estações do ano. Essas condições representam o mínimo necessário para que o trabalho de voluntários pudesse ser efetivamente exercido a fim de viabilizar a criação de uma robusta base de imagens que retratassem o contexto de objetos localizados em superfícies de rios da Amazônia.

Alternativamente ao cenário exposto acima, a fim de viabilizar, em um curto espaço de tempo, a criação de uma base de imagens que retratassem a região de interesse com as distintas condições de captura mencionadas acima, foram adotados métodos de busca para coleta de imagens a partir de repositórios volumosos de imagens e vídeos disponíveis em diversos sites na Internet. Entre as fontes utilizadas, destacam-se os seguintes sites: *Pixabay*, *Freepik*, *Pinterest*, *DepositPhotos*, *Getty Images*, *Google Imagens* e *Youtube*.

Todas essas fontes caracterizam-se pela pronta disponibilidade de compartilhamento de fotos, ilustrações, imagens vetoriais, e cenas de vídeo, de alta qualidade. Além disso, possuem sistemas inteligentes de indexação de modo que é possível buscar por fotos e vídeos relacionados com as palavras-chaves de interesse. Para encontrar imagens que retratassem superfície de rios da Amazônia, foram cruzadas palavras nos campos de buscas, tais como "rios", "amazônia", "river", "amazon", "barcos", "embarcações", "boats" e "vessels". Essas palavras foram as que trouxeram a maior quantidade de imagens e vídeos relevantes para o cenário de interesse da pesquisa. A seguir, são ilustradas duas imagens coletadas, fazendo menção à fonte utilizada.

Figura 8 – Exemplo de imagem coletada para composição da base de imagens. A figura ilustra uma embarcação navegando no rio Solimões.



Fonte: (PIXABAY, 2022).

Figura 9 – Exemplo de imagem coletada para composição da base de imagens. A figura ilustra casas flutuantes, um cenário rotineiro em rios da Amazônia.



Fonte: (FREEPIK, 2022).

O esforço para a criação da base de imagens foi dividido entre dois integrantes voluntários. Para que as imagens pudessem pertencer a base de imagens que estava sendo criada, elas deveriam atingir determinados critérios previamente estabelecidos e acordados entre os voluntários. Os critérios são listados a seguir.

- Cada imagem deve conter pelo menos um dos seguintes objetos de forma nítida (objeto que pode ser facilmente reconhecido sem o auxílio de técnicas de processamento digital de imagens): ponte, barco, lixo, rochas, plataforma, pessoa, canoa, casa ou flutuante;
- As imagens devem representar superfícies de rios da Amazônia, caracterizando imagens capturadas por perspectivas correspondentes a câmeras fixadas em embarcações;
- Buscar imagens que foram capturadas em distintas condições de iluminação ao longo do dia.

As classes de objetos listadas acima foram inspiradas a partir do trabalho realizado pelos autores em (ZHOU et al., 2021). Contudo, de um modo geral, as imagens pertencentes ao dataset *WSODD* não representam de forma satisfatória o cenário de objetos que são encontrados em superfícies dos rios navegáveis da Amazônia. Por exemplo, são encontradas muitas embarcações em formato de canoa nos leitos e margens dos rios dessa região. Entende-se por canoa uma embarcação leve, de pequeno porte, feita de uma só peça alongada, movida a remo, vela ou motor de rabeta, sendo considerado o meio de

transporte mais utilizado por ribeirinhos e indígenas nos rios da Amazônia. Esse tipo de embarcação não é bem representado no *dataset WSODD* e, em virtude disso, foi escolhida como uma classe de interesse a ser representada. Ainda exemplificando, muitos habitantes das comunidades indígenas e ribeirinhas residem em margens e leitos dos rios da Amazônia, caracterizando casas que flutuam sobre as águas, conhecidas popularmente como flutuantes. Em virtude da presença volumosa de flutuantes localizados nos leitos e margens dos rios, essa classe foi considerada como de interesse a ser detectada pelo modelo. Outra embarcação comum na Amazônia são as balsas - transporte fluvial de cargas muito utilizado por habitantes da região para fazer a travessia de rios juntamente com cargas maiores, tais como máquinas pesadas, equipamentos de construção, automóveis e entre outros. Essa classe de objetos foi escolhida para composição do *dataset* e foi rotulada como "plataforma" na lista de objetos citada mencionada anteriormente. Por último, outros objetos comumente localizados em superfícies fluviais foram escolhidos para criação do *dataset* e a Tabela 3, a seguir, fornece uma breve descrição sobre cada item.

Tabela 2 – Definição das categorias dos objetos escolhidos para detecção.

Categoria	Tradução	Descrição
boat	barco	Barcos de pequeno, médio ou grande porte típicos da Amazônia. Exemplos: voadeira, lanchas, obidense e etc.
building	edificações	Edificações localizadas nas margens e leitos nos rios da Amazônia. Exemplos: casas de madeira cobertas de palha que "flutuam" sobre as águas, restaurantes flutuantes, postos de abastecimento flutuantes e entre outros.
canoe	canoa	Embarcação leve a remo ou a vela, incluindo embarcações com motor de popa. Esse tipo de embarcação caracteriza barcos estreitos e leves.
person	pessoa	De maneira geral, pessoas foram consideradas alvos para ser detectadas pelo modelo.
plataform	plataforma	Embarcações que foram, por exemplo, projetadas para empurrar, puxar e rebocar outras embarcações Exemplos: balsas e rebocadores.
rubbish	lixo	Lixos dispersos nos leitos e margens dos rios da Amazônia.
ship	navio	Caracteriza grandes embarcações.

Fonte: (Elaborado pelo autor, 2022).

As imagens foram obtidas ora diretamente a partir dos repositórios de imagens mencionados acima, ora a partir de vídeos. Em relação as imagens obtidas a partir de vídeos, destaca-se como principal fonte de contribuição o *Youtube* - uma plataforma online internacionalmente reconhecida para compartilhamento de vídeos de diversos temas. Conforme explanado em (GONZALEZ; WOODS, 2009), sabe-se que um vídeo é uma sequência de quadros na qual cada quadro é uma imagem estática colorida. Por exemplo, considerando

um vídeo de 20 minutos, com resolução de 1080 x 720 pixels (HD) e taxa de quadros por segundo (FPS) de 30 FPS, a quantidade total de imagens estáticas que compõem esse vídeo é de:

$$N = \frac{size(bytes)}{framesize(bytes) * duration(seconds)} \quad (5)$$

em que:

N : quantidade de imagens estáticas.

$size(bytes)$: tamanho total do vídeo em bytes.

$framesize$: tamanho total de apenas uma única imagem RGB em bytes.

$duration$: duração do vídeo em segundos.

Considerando esse grande volume de imagens estáticas a partir de um único vídeo, foi realizado um trabalho exaustivo de analisar quadro por quadro imagens que representassem o contexto de objetos localizados em superfícies de diferentes rios navegáveis da Amazônia. Em virtude da disponibilidade de inúmeros vídeos relacionados ao contexto de interesse, foi possível coletar uma quantidade significativa de imagens para a composição do *dataset*. A Figura 10, a seguir, exemplifica uma imagem que representa um dos muitos quadros que compõem o vídeo "DE MANAUS A TEFÉ (Parte 2) — Caminhos da Amazônia - AMAZONAS 2022", disponível no *Youtube*. De maneira geral, esse vídeo disponibiliza imagens capturadas ao longo do trajeto fluvial realizada a partir da cidade de Manaus até o município Tefé, um percurso muito comum para os moradores e comerciantes dessa região.

Figura 10 – Quadro de imagem pertencente ao vídeo "DE MANAUS A TEFÉ (Parte 2) — Caminhos da Amazônia - AMAZONAS 2022".

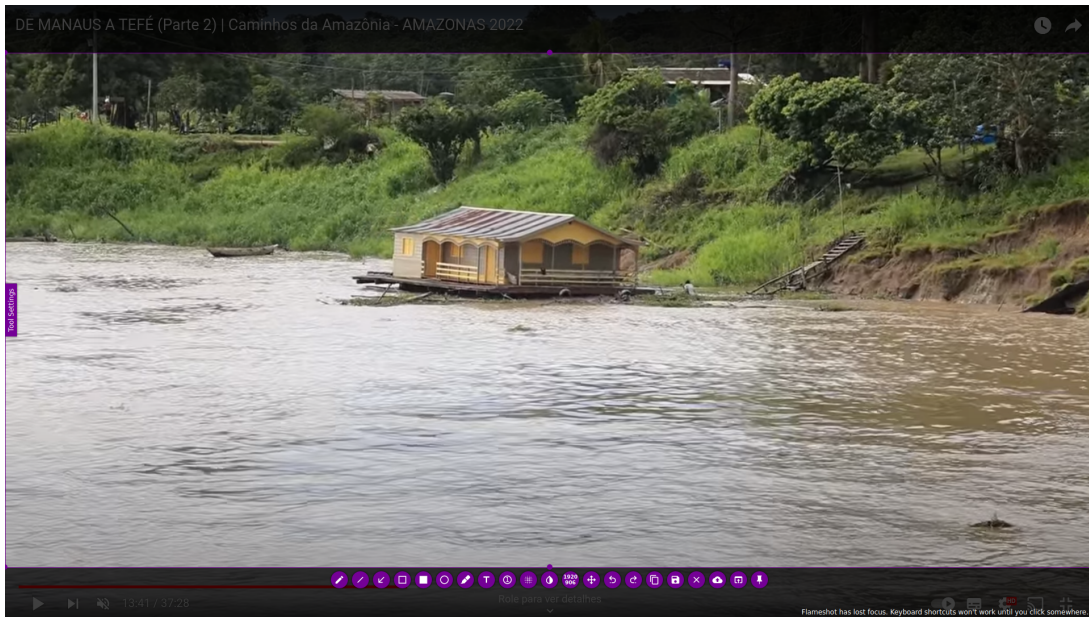


Fonte: (YOUTUBE, 2022).

Considerando ainda a Figura 10, observa-se que ela contém objetos de interesse e ilustra uma imagem que obedece aos critérios mencionados anteriormente. No caso específico

de quadro de imagens capturadas a partir de vídeos, é utilizado o software *Flameshot* - uma ferramenta simples e intuitiva de código livre e aberto, útil para capturar *screenshots* e, imediatamente, salvar a imagem capturada em um local especificado pelo usuário. A Figura 11 ilustra o procedimento utilizado para capturar um quadro de imagem e salvá-la em um diretório correspondente.

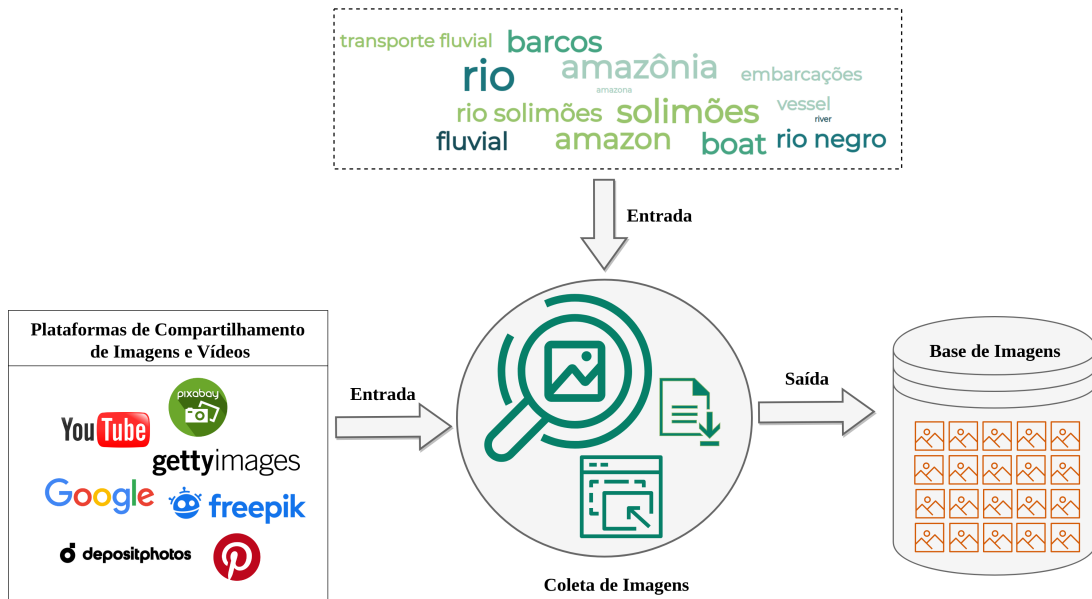
Figura 11 – Procedimento utilizado para capturar um quadro de imagem com a ferramenta *flameshot*.



Fonte: (Elaborado pelo autor, 2022).

Em síntese, os métodos empregados para coletar imagens publicamente disponíveis na *Internet* é ilustrado pela Figura 12. Ao todo, foram coletadas 485 imagens que representassem o contexto de objetos localizados em superfícies de rios da Amazônia. Conforme visto, foram utilizadas diferentes plataformas de compartilhamento de imagens e vídeos como fontes de dados para a criação da base de imagens. Após essa etapa de coleta de imagens, o próximo passo consistiu em anotar os objetos pertencentes em cada uma das 485 imagens, isto é, rotular explicitamente quais as categorias de objetos representadas em cada imagem.

Figura 12 – Visão geral do procedimento utilizado para coletar imagens.



Fonte: (Elaborado pelo autor, 2022).

3.3.2 Anotação de objetos

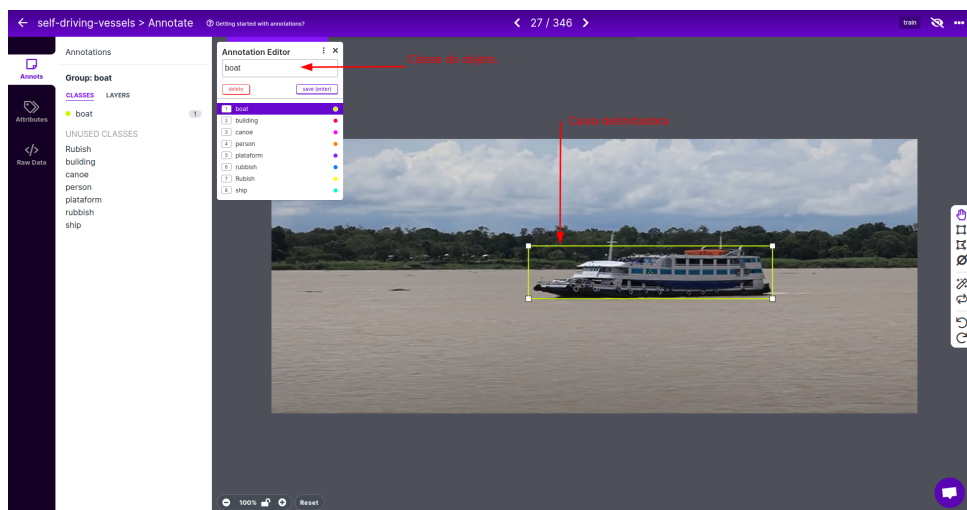
Um dos requisitos primordiais para modelos baseados em algoritmos de aprendizado profundo é a disponibilidade de um grande volume de dados rotulados (bases supervisionadas), isto é, dados em que a representação real da classe é conhecida. Especialmente no caso de dados de imagens, infelizmente, é comum não haver disponível, publicamente, uma base de imagens rotuladas para os propósitos de interesse de um determinado cenário de aplicação. Além disso, acontece que muitas vezes o processo de rotulagem é custoso, tanto do ponto de vista de tempo requerido, quanto do ponto de vista do investimento financeiro exigido. Ainda assim, o processo de rotulagem de uma base supervisionada é vital para a obtenção de modelos robustos de detecção de objetos, baseados em algoritmos de aprendizado profundo.

No contexto de detecção de objetos, anotar objetos em uma imagem significa prover os rótulos, isto é, as representações reais das classes dos objetos inseridos na imagem. Após a etapa de coleta de imagens, foram investidos esforços no processo de anotação dos objetos localizados nas superfícies de rios da Amazônia. Para auxiliar nesse processo de anotação de objetos, foram utilizados os recursos de software do *website Roboflow* - uma plataforma online que fornece suporte na implementação de soluções de visão computacional que se baseiam em modelos de detecção de objetos ou classificação de imagens. Um dos pontos positivos notado pelo autores no momento da decisão por essa plataforma diz respeito à possibilidade de trabalhar de forma cooperativa, ou seja, a possibilidade de dividir o trabalho de anotação dos objetos entre os voluntários de forma organizada, além de outras características úteis como, por exemplo, editar o nome das classes definidas, ter

acesso a indicadores do progresso do trabalho realizado por meio de *dashboards* (painéis de visualização), exportar a base de imagens com um formato específico e aplicar uma vasta variedade de métodos de pré-processamento a fim de expandir a quantidade de imagens disponíveis para o conjunto de treino (mais detalhes serão vistos adiante).

Para realizar o processo de anotação de objetos por meio dessa plataforma, primeiramente foi necessário transferir as imagens coletadas para o repositório de imagens online da plataforma. Após essa etapa, as imagens ficaram imediatamente disponíveis para serem anotadas. Para anotar um objeto, em síntese, é necessário circular o objeto de interesse por meio de uma "caixa delimitadora" (do inglês, *bounding box*) e, em seguida, especificar a categoria que o objeto pertence (rotulagem), conforme ilustrado pela Figura 13. Esse processo de anotação é repetitivo e foi realizado de forma cuidadosa a fim de circular, por meio de caixas delimitadoras, as posições dos diferentes objetos na imagem, bem como especificar adequadamente a categoria do objeto.

Figura 13 – Processo de anotação de um objeto com a ferramenta Roboflow.



Fonte: (Elaborado pelo autor, 2022).

Dentre os desafios encontrados durante esse processo de anotação de objetos, destacam-se a problemática da oclusão e múltiplas escalas de objetos. A problemática da oclusão trata-se de objetos que estão parcialmente ocultos em virtude da presença de outros objetos vizinhos, e também acontece quando por motivos diversos (nível da iluminação do ambiente, projeção de sombras, perspectiva da câmera e entre outros) um objeto é ocultado parcialmente ou totalmente. Essa problemática consiste em um desafio durante o processo de anotação de objetos, pois deve-se circular o objeto, mesmo que esteja parcialmente oculto, em sua totalidade, isto é, estimando onde estaria o contorno completo do objeto, conforme exemplificado a partir da ilustração da Figura 14.

Figura 14 – Exemplo de uma imagem com a presença de oclusão entre objetos.



Fonte: (Elaborado pelo autor, 2022).

Ao decorrer do processo de anotação, verificou-se que, em algumas imagens, determinados objetos apareciam em pequenas escalas, isto é, dimensões menores do objeto de interesse, enquanto que os mesmos objetos, em outras imagens, apareciam em escalas muito maiores, conforme ilustra a Figura 15. De toda forma, a diversificação de diferentes escalas (múltiplas escalas de objetos) contribuiu para a criação de uma base de imagens mais robusta, de modo que foi possível obter um modelo capaz de detectar objetos tanto em distâncias maiores (menores escalas), quanto em distâncias menores (maiores escalas). Em virtude disso, tais objetos inseridos nessa problemática também foram considerados ao decorrer do processo de anotação.

Figura 15 – Representação de um objeto em larga escala.



Fonte: (Elaborado pelo autor, 2022).

Figura 16 – Representação de um objeto em pequena escala.



Fonte: (Elaborado pelo autor, 2022).

Por último, destaca-se também o desafio em definir as categorias de interesse de modo que não houvesse sobreposição de objetos ambíguo. Em muitas imagens, os voluntários que participaram do processo de rotulagem do *dataset* relataram dificuldades em conseguir distinguir claramente em qual categoria o objeto anotado deveria pertencer. Por exemplo, muitas ocorrências de dúvidas aconteceram com embarcações, pois com frequência eram relatadas dificuldades em delimitar a fronteira entre o que separa um tipo de embarcação da outra, isto é, em que momento uma embarcação deixa de ser classificada como uma canoa e passa a ser classificada como barco, ou ainda, em que momento uma embarcação deixa de ser categorizada como barco e passa a ser classificada como navio. Dificuldades

como essa podem indicar que as categorias de objetos não estivessem bem definidas de modo que, em trabalhos futuros, tais categorias devem ser reformuladas por meio de uma revisão mais aprofundada.

Em síntese, o processo de coleta de imagens, em conjunto com processo de anotação de objetos, compuseram a primeira versão da base de imagens. Ao todo foram coletadas 485 imagens, com 2.035 objetos anotados. A Tabela 3, a seguir, fornece um resumo da quantidade de objetos representados por cada categoria.

Tabela 3 – Dados da quantidade de objetos representados por categoria na base de imagens criada.

Categoria	Quantidade
boat	660
building	248
canoe	468
person	523
plataform	86
rubbish	18
ship	32

Fonte: (Elaborado pelo autor, 2022).

Considerando as informações disponibilizadas acima, verificou-se que a quantidade de imagens coletadas foi insuficiente para o treinamento de um modulo robusto de detecção de objetos localizados nas superfícies de rios da Amazônia. Para contornar esse problema, foram aplicados métodos de expansão de imagens (do inglês, *Data Augmentation*), conforme será visto a seguir.

3.3.3 Técnicas de pré-processamento de imagens

Após a finalização do processo de coleta de imagens e rotulagem de objetos, a primeira versão do *dataset* de imagens foi composta por 485 imagens. De maneira geral, as imagens coletadas foram caracterizadas por possuírem os 3 canais de cores RGB (coloridas) e não possuírem uma dimensão fixa, isto é, variaram em diferentes dimensões de pixels. Considerando isso, dois principais métodos de pré-processamento foram aplicados ao *dataset*: redimensionamento das imagens para uma dimensão fixa e técnicas de aumento da base de imagens.

Conforme mencionado acima, a primeira versão da base de imagens foi composta por uma quantidade de imagens, a priori, insuficiente para modelos de detecção de objetos conseguirem obter uma notável performance frente a diferentes classes de objetos. Para

lidar com esse cenário, foi proposta a utilização de métodos para aumentar a quantidade de imagens disponíveis para o conjunto de treinamento.

Conforme visto na Seção 1, técnicas para aumentar a base de imagens podem ser empregadas para, artificialmente, expandir o tamanho do conjunto de treino por meio da aplicação de diferentes filtros nas imagens originais. Dessa forma, o modelo de detecção de objetos localizados nas superfícies de rios da Amazônia ficou exposto a uma quantidade maior de imagens, melhorando notavelmente sua capacidade de generalização ao detectar diferentes categorias de objetos em novas imagens.

A fim de aumentar a base de imagens com técnicas de expansão, a plataforma *Robo-flow*, apresentada anteriormente, teve fundamental relevância, pois provisionou, de forma rápida e prática, acesso a diferentes técnicas de processamento de imagens para aumentar o conjunto de treino. A Tabela 4 fornece uma visão geral das transformações empregadas nas imagens, bem como a faixa de variação para cada uma.

Tabela 4 – Transformações aplicadas nas imagens para o processo de aumento da base imagens de treinamento.

Técnica	Tradução	Especificação
<i>flip</i>	inversão	Horizontal
<i>crop</i>	recorte	0% Zoom Mínimo - 50% Zoom Máximo
<i>rotation</i>	rotação	Entre -9° e $+9^\circ$
<i>shear</i>	cisalhamento	Entre $\pm 9^\circ$ Horizontal e $\pm 9^\circ$ Vertical
<i>grayscale</i>	escala de cinzas	Aplicado para 25 % das imagens
<i>saturation</i>	saturação	Entre -25% e +25%
<i>brightness</i>	brilho	Entre -8% e +8%
<i>blur</i>	desfoque	Dimensão do kernel de até 5.5 pixels
<i>noise</i>	ruído	Até 3% dos pixels são afetados
<i>mosaic</i>	mosaico	Efeito de mosaico é aplicado
<i>bounding box: rotation</i>	caixa delimitante: rotação	Entre -7° and $+7^\circ$
<i>bounding box: blur</i>	caixa delimitante: desfoque	Dimensão do kernel de até 5 pixels

Fonte: (Elaborado pelo autor, 2022).

Após a aplicação dos filtros listados na Tabela 4, foi possível produzir um aumento de 485 para 1305 imagens, representando um aumento significativo de quase três vezes mais imagens disponíveis no conjunto de treino. Tendo sido finalizado essa etapa de pré-processamento, esforços foram investidos no treinamento do modelo de detecção de objetos.

3.3.4 Validação do modelo

No que tange a validação do modelo, foi utilizado o método *holdout* para a validação cruzada do experimento. Contudo, diferentemente da validação descrita na Seção 3.2.2, foi utilizado uma proporção diferente de imagens disponíveis nos conjuntos de treino e

validação, isto é, 90 % do total das imagens foram destinadas ao conjunto de treino, totalizando 1305 imagens, e os 10 % de imagens restantes foram destinadas ao conjunto de validação, totalizando 50 imagens para esse conjunto. Ressalta-se que a seleção delas, em ambos os conjuntos, foi realizada de forma aleatória, mantendo uma distribuição representativa das classes em cada conjunto. Especialmente no que diz respeito ao conjunto de treino, as 1305 imagens foram geradas a partir de técnicas de expansão de modo a produzir 3 vezes mais imagens disponíveis no conjunto de treino, conforme detalhado na seção anterior.

3.3.5 Treinamento do detector de objetos

Tendo sido criada a base de imagens correspondentes a objetos localizados em superfícies de rios da Amazônia, esforços foram direcionados ao treinamento do modelo responsável por realizar a detecção de objetos. De maneira geral, a etapa de treinamento foi implementada de forma muito semelhante aos procedimentos e materiais utilizados na seção 3.2, sendo também escolhido o algoritmo *YOLOv5* como método responsável pelo processo de detecção de objetos. Apesar da semelhança entre as abordagens, houve uma diferença crucial, pois os pesos do modelo treinado a partir das abordagens definidas na seção 3.2 foram reutilizados como ponto de partida (inicializadores) para o treinamento do modelo que foi implementado nesta seção, caracterizando uma abordagem baseada em métodos aprendizados por transferência (do inglês, *transfer learning*).

Conforme visto na Seção 1, métodos de aprendizados por transferência consistem em utilizar características (pesos) de um outro problema, e aproveitá-las para um novo e similar problema. Sob essa ótica, considerando o modelo que foi treinado para a base de imagens *WSODD*, observa-se que este modelo, certamente, aprendeu a detectar traços e características profundas de interesse dos objetos localizados em superfícies de regiões representadas pela predominância de água. Dessa forma, foi possível reaproveitar os pesos do modelo anterior de modo a contribuir no treinamento do modelo responsável pela detecção de objetos localizados em superfícies de rios da Amazônia que, ressalta-se, representa um contexto de aplicação análogo ao representado pelas imagens que compõe o *dataset WSODD*.

Seguindo a abordagem proposta acima, foram utilizadas 1305 imagens para o treinamento do modelo. Conforme visto na Seção 3.3.4, 50 imagens foram utilizadas para o conjunto de validação. Os resultados serão apresentados e discutidos na próxima seção. A seguir, são descritos detalhes de implementação utilizados ao decorrer da etapa de treinamento do modelo de detecção de objetos.

3.3.6 Detalhes de implementação

Em relação aos detalhes de implementação, para realizar os experimentos com o método *YOLOv5*, foi utilizado para o modelo treinado a partir da base de imagens *WSODD* uma estação de trabalho (computador) com as seguintes configurações:

- Sistema operacional Ubuntu 20.04;
- Processador Intel Core i7-7700HQ de 7^a Geração;
- Memória RAM 16GB DDR4;
- Armazenamento HD de 1TB 5400 RPM;
- Placa de vídeo dedicada NVIDIA GeForce GTX 1060 com 6GB gDDR5.

No segundo momento, para o modelo treinado a partir da base de imagens regional, foi utilizada uma estação de trabalho com as seguintes configurações:

- Sistema operacional Ubuntu 22.04;
- Processador Intel Core i7-12700H x 20;
- Memória RAM 32,0 GiB DDR4;
- Armazenamento SSD de 512 GB;
- Placa de vídeo dedicada NVIDIA GeForce RTX 3050 com 4 GB.

A escolha entre a primeira ou a segunda configuração não foi um fator que influenciou na performance do modelo após o treinamento. Contudo, a segunda configuração utilizada viabilizou um treinamento mais rápido (menor custo de tempo de processamento) do modelo em virtude de ter sido utilizado uma GPU mais robusta, com maior entrega de processamento paralelo.

Em ambos os modelos, as instâncias de treinamento e validação foram executadas em contêineres, isto é, em ambientes de desenvolvimento isolados. Para provisioná-los, foi utilizado um software de código aberto especializado em virtualização de ambientes de desenvolvimento por meio de contêineres. Assim, foi implementado, em um curto espaço de tempo, as etapas necessárias para o treinamento e avaliação dos modelos com o método *YOLOv5*, visto que todas as dependências de software requisitadas por esse método já estavam empacotadas e disponibilizadas por uma imagem pronta para uso.⁴

Por fim, para os hiper-parâmetros utilizados na arquitetura convolucional do detector *YOLOv5*, foi definido uma taxa de aprendizado de 0.01, momento igual à 0.937 e decaimento de peso de 0.0005. Em seguida, definiu-se o tamanho de lote igual à 16, com número épocas igual à 300. Além disso, para a otimização da função de perda, foi escolhido o otimizador *SGD*⁵ (*Stochastic Gradient Descent*). Observa-se que, como etapa de

⁴Referências sobre o uso do *YOLOv5* com *Docker*, consultar: <https://docs.ultralytics.com/environments/Docker-Quickstart/>

⁵Mais informações em: https://en.wikipedia.org/wiki/Stochastic_gradient_descent

preprocessamento, as imagens de entrada foram redimensionadas para 416 x 416 pixels. As escolhas desses hiper-parâmetros foram baseadas por meio da observação empírica, isto é, tentativa e erro, além de também se basear nos valores padrões utilizados pelo método *YOLOv4* na base de imagens *PASCAL VOC*.

3.3.7 Integração do modelo em um dispositivo embarcado

A viabilização de modelos de detecção de objetos para o contexto de embarcações autônomas tem como um requisito importante a utilização de computadores de borda com suporte a GPUs (do inglês, edge devices). Computadores de borda tornam possível, por exemplo, o processamento de imagens em tempo real na borda da rede, isto é, o mais próximo possível da fonte de dados, viabilizando serviços de detecção de objetos com menor latência, mais confiáveis e mais flexíveis. No contexto de embarcações autônomas, essas características são essenciais à medida que eliminam a necessidade de enviar grandes volumes de informações para um datacenter central a partir de locais remotos e de difícil acesso a banda larga móvel.

Tais computadores são caracterizados por possuírem dimensões inferiores comparadas a um computador comum, porém apresentam desempenhos satisfatórios para aplicações que requerem a utilização paralela de múltiplas redes neurais convolucionais, tais como no caso de detecção de objetos para o contexto de embarcações autônomas. Por meio de um computador de borda, é possível embarcar um modelo nele de modo que fique responsável apenas pela detecção de objetos, sem que seja necessário um computador robusto para desempenhar inferências. De fato, normalmente, computadores robustos integrados com GPUs de alta performance são utilizados apenas para a etapa de treinamento de modelos, pois tal etapa exige mais recursos de software e hardware para ser realizada. Após a etapa de treinamento, os pesos dos modelos são salvos em formatos específicos de modo que possam ser reaproveitados em um computador de borda, fazendo com que este não precise de um hardware para treinamento de modelos, mas sim apenas para realização de inferências.

Com a finalização da etapa de treinamento e validação do modelo para detecção de objetos localizados em superfícies de rios da Amazônia, esforços foram investidos para realizar a integração do modelo em um computador de borda. Assim, foi definido o Kit de Desenvolvimento *Jetson Nano* (Figura 17) como computador de borda para realizar a integração proposta. Essa plataforma embarcada é caracterizada por provisionar recursos de hardware e software orientado para rápida prototipação de aplicações que envolvam, por exemplo, executar modelos de visão computacional ou inteligência artificial.

Figura 17 – Ilustração do Kit de Desenvolvimento *Jetson Nano*.

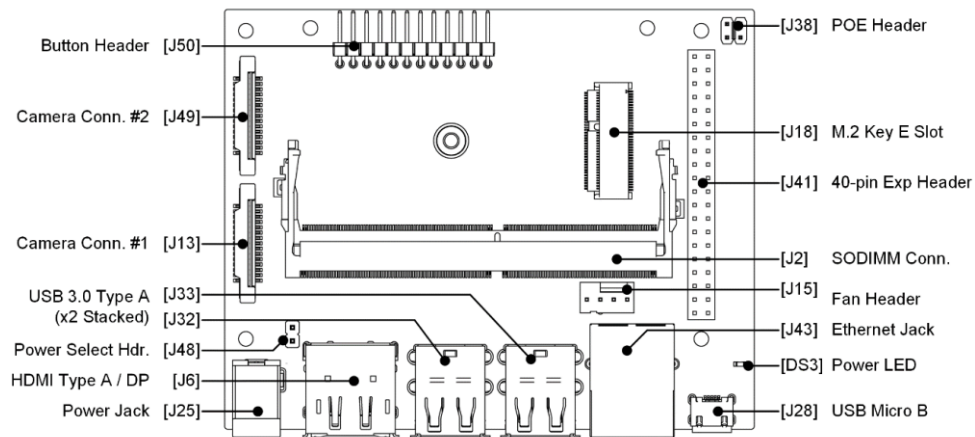


Fonte: (NVIDIA, 2022).

A Figura 18 ilustra as principais interfaces do Kit de Desenvolvimento *Jetson Nano*. Dentre as suas especificações técnicas, destacam-se as seguintes configurações:

- Sistema operacional Jetson Linux;
- Processador ARM A57 com 4 núcleos e frequência de 1.43 GHz;
- Memória RAM LPDDR4 de 4GB (64 bits) E 25.6 GB/s;
- Capacidade para processar vídeos com resolução de até 4K em taxas de 30 quadros por segundo (FPS);
- Armazenamento via cartão de memória microSD
- Placa de vídeo integrada (GPU) com arquitetura Maxwell de 128 bits.
- Suporte aos protocolos de comunicação GPIO, I2C, I2S, SPI e UART.

Figura 18 – Interfaces do Kit de Desenvolvimento *Jetson Nano*.

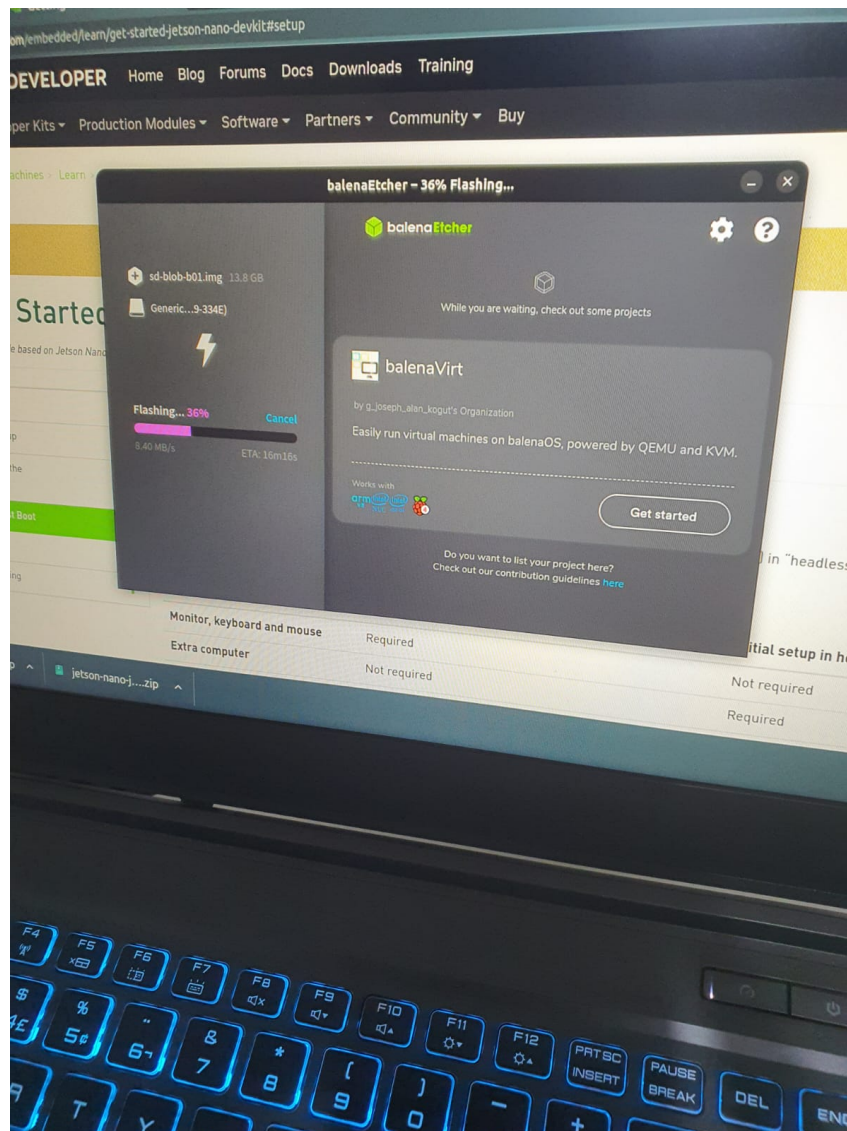


Fonte: (NVIDIA, 2022).

A fim de realizar a integração entre o modelo treinado e a plataforma embarcada *Jetson Nano*, foi necessário realizar uma sequência de configurações. O primeiro passo foi gravar uma imagem do sistema operacional Jetson em um cartão de memória microSD de 64 GB para poder inicializá-lo, conforme ilustra a Figura 20. Esse sistema operacional faz parte da pilha de softwares disponibilizados pelo SDK (do inglês, Software Develop Kit) JetPack - um conjunto de ferramentas Nvidia que dão suporte ao desenvolvimento de aplicações de visão computacional e inteligência artificial. Para essa etapa, foi utilizado a ferramenta gratuita Balena Etcher⁶.

⁶Referências sobre o uso do *software Balena Etcher* consultar: <https://www.balena.io/etcher/>.

Figura 19 – Processo de gravação da imagem do sistema operacional Jetson Linux em um cartão de memória com a ferramenta Etcher.



Fonte: (Elaborado pelo autor, 2022).

Após a finalização do primeiro passo, a segunda etapa consistiu-se em realizar a primeira inicialização da plataforma *Jetson Nano*. Para isso, foi necessário conectar periféricos, tais como monitor, teclado, mouse, cabo HDMI, cabo de rede Ethernet e câmera a fim de viabilizar a configuração do dispositivo de forma interativa. A Figura 20 ilustra os periféricos conectado, bem como a tela de inicialização.

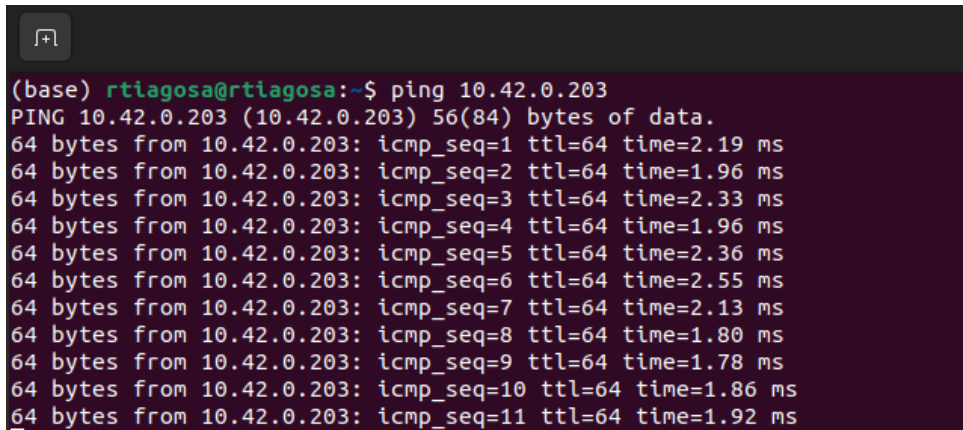
Figura 20 – Periféricos conectados na plataforma *Jetson Nano* e, ao lado esquerdo, exibição do sistema operacional através de um monitor conectado via cabo HDMI.



Fonte: (Elaborado pelo autor, 2022).

Em seguida, como terceira etapa, foi necessário estabelecer a conectividade entre o dispositivo e a rede local (conhecido também como LAN - Local Area Network) por meio de uma conexão Ethernet com o roteador local, sendo este responsável por estabelecer a comunicação entre a rede local e à Internet. Isso tornou possível a troca de dados entre a máquina *host* (estação desenvolvimento) e a máquina *target* (plataforma embarcada *Jetson Nano*). A Figura 21 ilustra, por meio da execução do comando *ping*, um teste comum de conectividade entre a estação de desenvolvimento e a plataforma embarcada *Jetson Nano*.

Figura 21 – Teste de conexão com o dispositivo embarcado através do comando ping.



```
(base) rtiagosa@rtiagosa:~$ ping 10.42.0.203
PING 10.42.0.203 (10.42.0.203) 56(84) bytes of data.
64 bytes from 10.42.0.203: icmp_seq=1 ttl=64 time=2.19 ms
64 bytes from 10.42.0.203: icmp_seq=2 ttl=64 time=1.96 ms
64 bytes from 10.42.0.203: icmp_seq=3 ttl=64 time=2.33 ms
64 bytes from 10.42.0.203: icmp_seq=4 ttl=64 time=1.96 ms
64 bytes from 10.42.0.203: icmp_seq=5 ttl=64 time=2.36 ms
64 bytes from 10.42.0.203: icmp_seq=6 ttl=64 time=2.55 ms
64 bytes from 10.42.0.203: icmp_seq=7 ttl=64 time=2.13 ms
64 bytes from 10.42.0.203: icmp_seq=8 ttl=64 time=1.80 ms
64 bytes from 10.42.0.203: icmp_seq=9 ttl=64 time=1.78 ms
64 bytes from 10.42.0.203: icmp_seq=10 ttl=64 time=1.86 ms
64 bytes from 10.42.0.203: icmp_seq=11 ttl=64 time=1.92 ms
```

Fonte: (Elaborado pelo autor, 2022).

Estabelecido a comunicação entre a estação de desenvolvimento e a plataforma embarcada, foi possível ter acesso remoto ao *Jetson Nano* por meio do serviço *SSH* (*Secure Shell*)⁷, permitindo a transferências entre *scripts*, dados e comandos de execução.

Com o objetivo de integrar o modelo com o dispositivo embarcado, a quarta etapa consistiu-se em configurar um contêiner (NVIDIA Container Runtime⁸) com todos os recursos de software necessários para executar o modelo de detecção de objetos baseados no algoritmo YOLOv5, incluindo uso da GPU para inferência. Conforme mencionado anteriormente, contêineres são como ambientes de desenvolvimento isolados com todas as dependências de softwares pré-configuradas, isto é, prontas para uso. A utilização deles viabiliza a prototipação da aplicação em um intervalo de tempo menor.

Após a finalização de todas as etapas acima, a quinta etapa consistiu-se em conectar uma câmera USB (Webcam HD Logitech C270) na plataforma embarcada e executar o contêiner de modo que ele pudesse realizar as seguintes operações: (i) prover um ambiente isolado com todas as dependências de softwares necessários para carregar o modelo, incluindo recursos da GPU; (ii) executar o *script* que carrega o modelo treinado para a detecção de objetos localizados em superfícies de rios da Amazônia;

Por último, a partir do momento em que o modelo integrado na plataforma embarcada *Jetson Nano* iniciou o processo de detecção de objetos a partir da captura de imagens com a câmera USB, vídeos correspondentes a objetos localizados em superfícies de rios da Amazônia foram transmitidos em um monitor de modo que pudessem simular, em tempo real de execução, a captura de uma sequência imagens em um cenário real, viabilizando a extração de métricas de desempenho da inferência em taxas de quadros por segundo (FPS).

⁷Mais informações sobre o serviço SSH, consultar: https://pt.wikipedia.org/wiki/Secure_Shell

⁸Referências sobre como utilizar, consultar: <https://developer.nvidia.com/nvidia-container-runtime>

Na próxima seção, serão apresentados os resultados, avaliações e discussões obtidos para cada um dos tópicos descritos nessa seção.

4 RESULTADOS

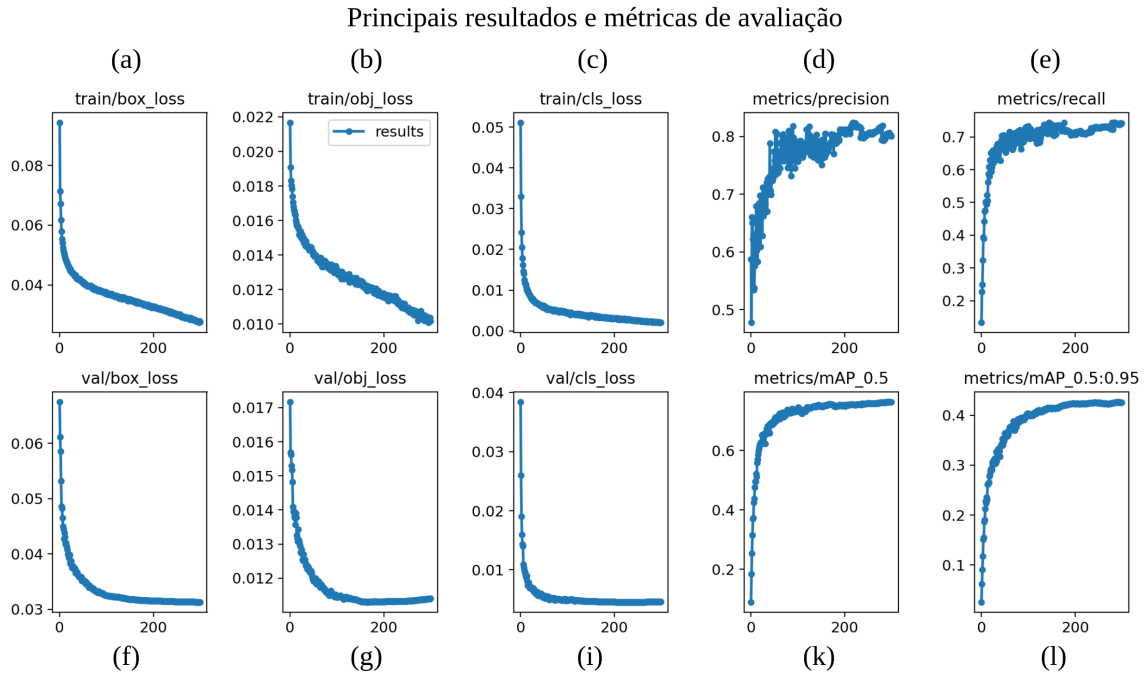
Essa seção disponibiliza os resultados e discussões que dizem respeito aos dois modelos treinados a partir do algoritmo YOLOv5. A seção é dividida em três partes: (i) discussões em torno das métricas de avaliação de performance do modelo treinado a partir da base de imagem WSODD, (ii) comparações entre a performance obtida nesse modelo com a performance de 17 outros detectores apresentados em (ZHOU et al., 2021) e, por último, (iii) discussões em torno das métricas de avaliação de performance do modelo treinado a partir da base de imagens regionais.

4.1 Avaliação de performance do modelo treinado a partir da base WSODD

4.1.1 Análise da função de perda

A curva da função de perda fornece um "retrato" do processo de treinamento do modelo e permite observar se o modelo está entrando em um processo de *overfitting* ou *underfitting*. Figura 22-a, Figura 22-b e Figura 22-c ilustram essa curva para três diferentes variáveis, isto é, *box_loss*, *obj_loss* e *cls_loss*. Observa-se que, à medida que o número de épocas aumenta, há uma convergência relativamente rápida no sentido da redução do erro. Por fim, a Figura 22-f, Figura 22-g e Figura 22-i ilustram o decaimento do erro no conjunto de validação para as mesmas três variáveis mencionadas acima.

Figura 22 – Visão geral dos resultados do treinamento do modelo baseado no detector YOLOv5. Os gráficos (a), (b), (c), (f), (g) e (i) ilustram a curva da função de perda para 3 diferentes variáveis, considerando tanto o conjunto de treino (*train*), quanto o conjunto de validação. A variação das métricas de precisão (*precision*) e revocação (*recall*) são ilustrados pelos gráficos (d) e (e). Por fim, as curvas ilustradas nos gráficos (k) e (l) fornecem resultados de mAP do modelo.



Fonte: (Elaborado pelo autor, 2022).

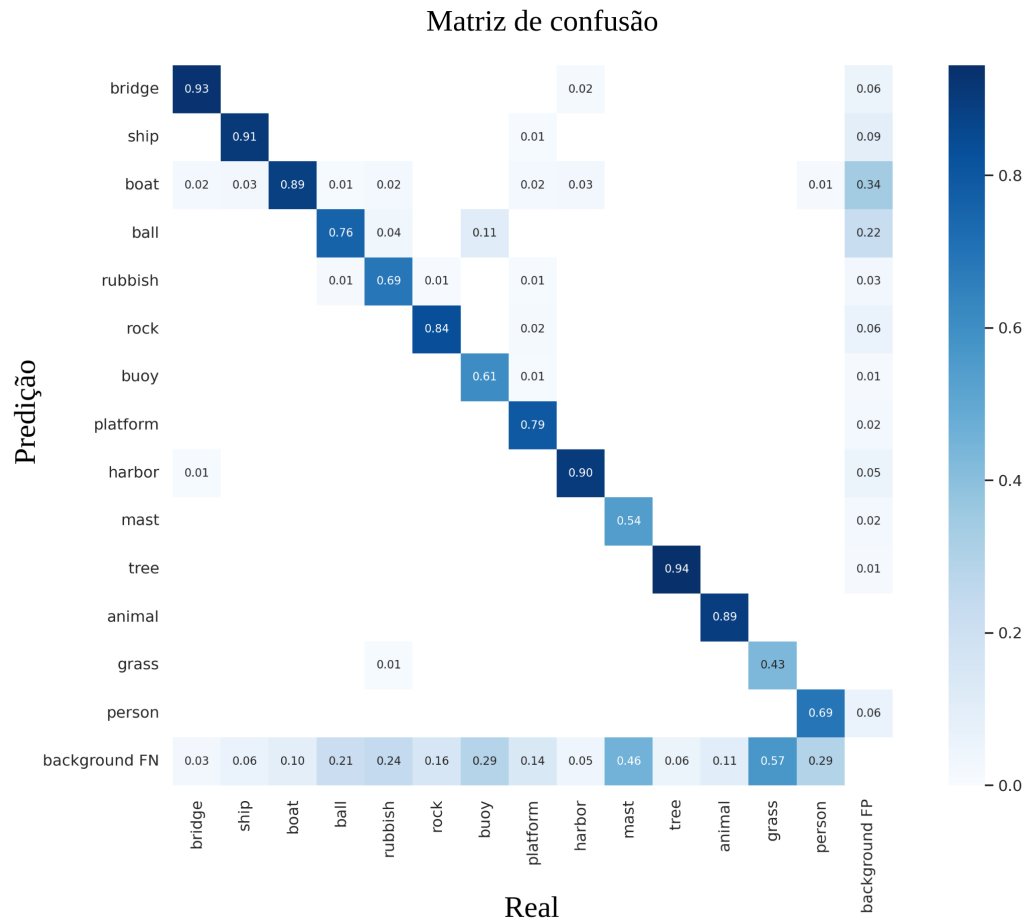
4.1.2 Análise das métricas precisão e revocação

A precisão e revocação são métricas de avaliação de performance comuns em modelos de detecção de objetos, pois conseguem medir a qualidade das classificações em termos de verdadeiros positivos (TP), falsos positivos (FP) e falsos negativos (FN). Figura 22-d e Figura 22-e ilustram como as métricas de precisão e revocação evoluíram ao longo do treinamento.

Ainda, observa-se que o modelo conseguiu aprender a detectar, com notável performance, objetos de diferentes categorias, pois houveram incrementos positivos na precisão e revocação à medida em que os falsos positivos e falsos negativos foram reduzidos.

A Figura 23 sintetiza o resultado das classificações para cada categoria por meio de uma matriz de confusão. Por meio dela, observa-se que as categorias que tiveram a maior porcentagem de instâncias classificadas erroneamente foram as classes *grass* (grama) e *mast* (mastro). A razão disso pode ser em virtude da baixa quantidade de imagens e instância que representam essas classes no conjunto de treino.

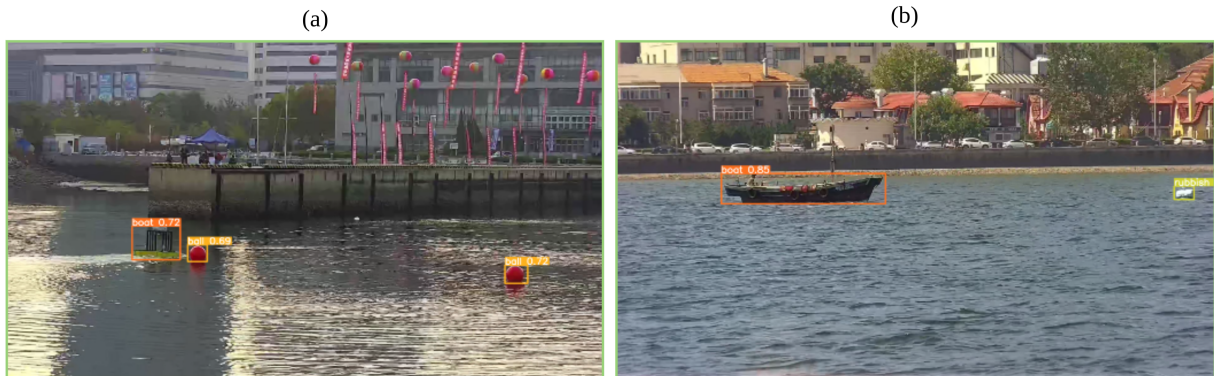
Figura 23 – Matriz de confusão das classificações obtidas pelo modelo.



Fonte: (Elaborado pelo autor, 2022).

Por fim, a Figura 31 ilustra imagens pertencentes ao conjunto de teste que foram submetidas ao processo de inferência para detecção de objetos localizados em superfície de água.

Figura 24 – Exemplos de objetos detectados após o treinamento do modelo. Na figura (a) nota-se a presença de caixas delimitadoras em volta de um barco e de uma bola (do inglês, *ball*). Na figura (b), são detectados dois objetos: um barco (*boat*) e um entulho (*rubbish*).



Fonte: (Elaborado pelo autor, 2022).

4.2 Comparações entre performances de modelos similares

Ao comparar métodos tradicionais de detecção de objetos com métodos baseados em arquiteturas convolucionais profundas, verifica-se que estes últimos têm alcançado uma maturidade significativa em um curto espaço de tempo, pois eles possuem capacidade de aprendizado muito superior aos métodos clássicos. Isso é evidenciado pelos resultados obtidos e disponibilizados na Tabela 5, onde são avaliados diversos modelos de detecção de objetos em função da métrica mAP. Todos os modelos apresentados nessa tabela foram treinados a partir da base de imagens *WSODD*, utilizando o mesmo critério de validação mencionado na seção 4.2. A seguir, são discutidos os resultados alcançados comparando-os com os resultados obtidos a partir do algoritmo *YOLOv5*.

Tabela 5 – Comparação entre a performance de 17 distintos métodos de detecção de objetos.

Método	FPS	mAP (%)
DPM	42.15	21.9
RANSAC-SVM	43.51	27.1
Faster R-CNN	19.42	32.3
Mask R-CNN	17.79	35.7
Cascade R-CNN	29.56	41.1
TridentNet	10.16	62.2
EfficientDet	30.83	31.3
RetinaNet	33.84	27.9
RFBNet	44.61	35.7
M2Det	40.63	39.3
SSD	43.02	41.5
CenterNet	43.42	53.5
Yolov3	45.34	56.1
Yolov3-2SMA	50.46	56.9
Yolov4	46.25	57.2
ShipYolo	49.81	58.4
CRB-Net	43.76	65.0
Yolov5	43.55	76.3

Fonte: (Elaborado pelo autor, 2022)

Em destaque, foi adicionado a contribuição deste trabalho, onde os resultados alcançados com o detector *YOLOv5* foram, em termos de mAP, significativamente superiores aos resultados dos outros 17 modelos apresentados em (ZHOU et al., 2021). Assim, verificou-se que o modelo obtido a partir do método *YOLOv5*, até o momento da escrita deste trabalho, obteve uma performance em estado da arte para detecção de objetos localizados em superfície de água, tendo como dataset de *benchmark* a base de imagens *WSODD*.

Os dois primeiros modelos (*DPM* e *RANSAC-SVM*) correspondem métodos de detecção de objetos baseados em técnicas clássicas de aprendizado de máquina. Observa-se um mAP baixo comparados os outros modelos. Todos os outros modelos restantes apresentados foram implementados baseadas em técnicas de aprendizado profundo. Em especial, os métodos *Yolov3-2SMA* e *ShipYolo* foram desenvolvidos para o cenário de detecção de objetos localizados em superfície de água. Em destaque, foi adicionado a contribuição deste trabalho em que o modelo treinado a partir do método *YOLOv5* obteve um mAP

de 76.3 %, considerando a base de imagens *WSODD*. Desse modo, a partir dos resultados obtidos, verificou-se que esse modelo alcançou notável capacidade de generalização para objetos localizados em superfície de água.

O ganho de performance do método *YOLOv5* em comparação aos outros modelos apresentados pode ser justificado pela utilização de técnicas de aumento de imagens que são aplicadas em lote durante a etapa de treinamento, e também pela robustez do algoritmo *YOLOv5*, corroborando para o desenvolvimento de um modelo que generaliza melhor frente a diferentes condições de capturas das imagens disponíveis no *dataset WSODD*.

Apesar da satisfatória capacidade de detecção de objetos para o contexto das imagens pertencentes ao *dataset WSODD*, observou-se que o modelo treinado teve notável dificuldade para detectar corretamente objetos localizados em cenários que representassem o contexto de rios da Amazônia, identificando um problema e, ao mesmo tempo, uma oportunidade para treinar um novo modelo capaz de contornar esse problema. A Figura 25 ilustra uma imagem pertencente à base de imagens *WSODD* e, em seguida, a Figura 26 ilustra uma imagem pertencente a base criada, retratando paisagens típicas de rios da Amazônia.

Figura 25 – Imagens pertencentes ao *dataset WSODD*, retratando uma superfície marítima.



Fonte: (Elaborado pelo autor, 2022).

Figura 26 – Imagens pertencentes ao *dataset* regional criado, representando uma superfície típica dos rios da Amazônia.



Fonte: (Elaborado pelo autor, 2022).

Conforme é possível observar a partir das imagens ilustradas acima, existem notórias diferenças entre o contexto das duas imagens. Pela Figura 25, observa-se a presença de construções de prédios nas margens da superfície marítima, enquanto que, pela Figura 26, observa-se a presença de edificações localizadas não apenas nas margens, mas também no leito do rio. Além disso, a superfície pela qual os barcos navegam apresenta aspectos distintos de tonalidade. Outros exemplos também evidenciam a diferença entre as duas bases de imagens, especialmente no que diz respeito a presença de muitos flutuantes, tipos diferentes de embarcações e alta densidade de vegetação nas margens dos rios. A Figura 27, a seguir, ilustra alguns erros de detecção em imagens correspondentes às superfícies de rios da Amazônia.

Figura 27 – Objetos detectados e classificados pelo modelo.



Fonte: (Elaborado pelo autor, 2022).

Considerando a Figura 27, observa-se alguns quadros de imagens demarcados com o símbolo "X" em cor vermelha, representando objetos detectados que foram classificados de forma errônea ou que, nem ao menos, foram detectados pelo modelo. Por exemplo, no quadro localizado na quarta linha e segunda coluna, verifica-se que a edificação flutuante foi classificada sendo um objeto pertencente à categoria "barco" com uma confiança de 80 %. Considerando as dificuldades do modelo para detectar objetos localizados nas superfícies de rios da Amazônia, foi proposto a criação de uma base de imagens regional seguida de um novo modelo, treinado a partir dessa base de imagens criada. A seguir, serão apresentados os resultados das métricas de avaliação de performance desse novo modelo treinado a partir de uma base de imagens regional.

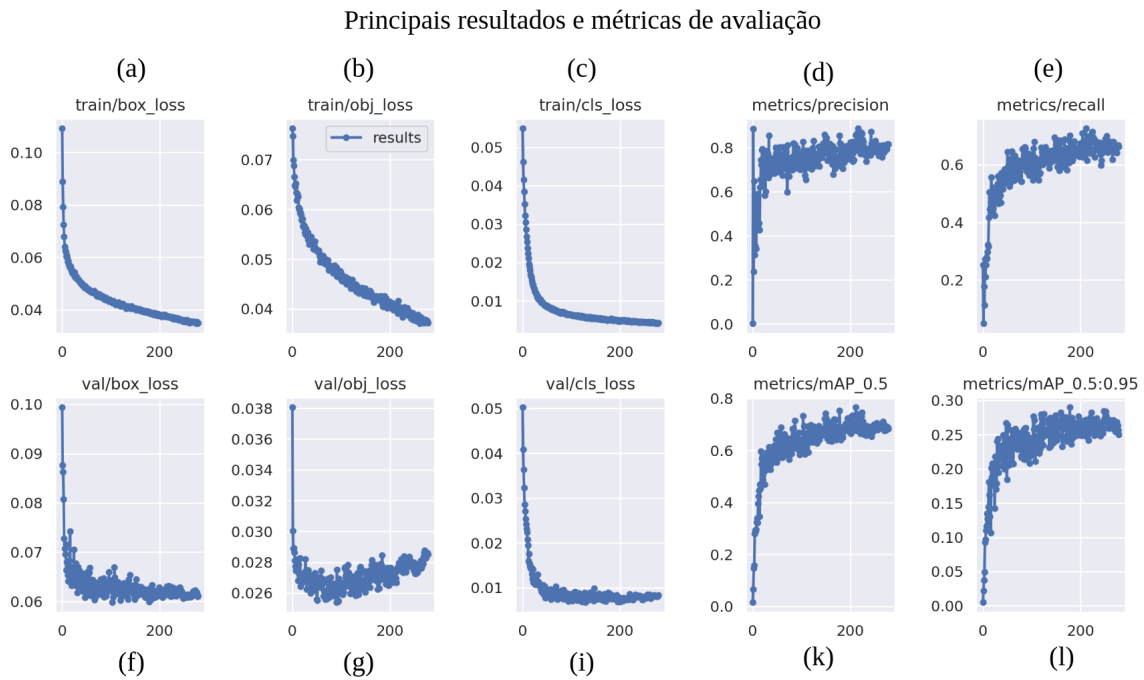
4.3 Avaliação de performance do modelo treinado a partir da base de imagens WSOD-ARD

4.3.1 Análise da função de perda

Para modelos de detecção de objetos baseados no algoritmo *YOLOv5*, conforme visto anteriormente, a função de perda costuma ser analisada sob três principais parâmetros: coordenadas da caixa delimitadora (*box_loss*), predição da caixa delimitadora (*obj_loss*) e classificação da categoria do objeto (*cls_loss*). Todos esses aspectos tem como comum objetivo a convergência para o mínimo erro possível a partir da análise da função de perda. Figura 22-a, Figura 22-b e Figura 22-c ilustram os parâmetros mencionados acima para

a etapa de treino, sendo possível observar que, conforme o número de épocas aumenta ao longo do treinamento, o erro tende a convergir cada vez mais para próximo de zero. Contudo, ao analisar paralelamente a Figura 22-f, Figura 22-g e Figura 22-l, observa-se que, a partir de um determinado número de épocas, o erro, ao invés de convergir para o mínimo global, apresenta um comportamento inverso, isto é, observa-se que o erro aumenta ao invés de diminuir. Em síntese, a análise dos três principais parâmetros mencionados permitem entender se o modelo está apresentando indícios de boa performance ou não.

Figura 28 – Visão geral das métricas de performance do modelo treinado a partir da base de imagens regional. Os gráficos (a), (b), (c), (f), (g) e (i) ilustram a curva da função de perda para 3 diferentes variáveis, considerando tanto o conjunto de treino (*train*), quanto o conjunto de validação. A variação das métricas precisão (*precision*) e revocação (*recall*) são ilustrados pelos gráficos (d) e (e). Por fim, as curvas ilustradas nos gráficos (k) e (l) fornecem resultados de mAP do modelo.



Fonte: (Elaborado pelo autor, 2022).

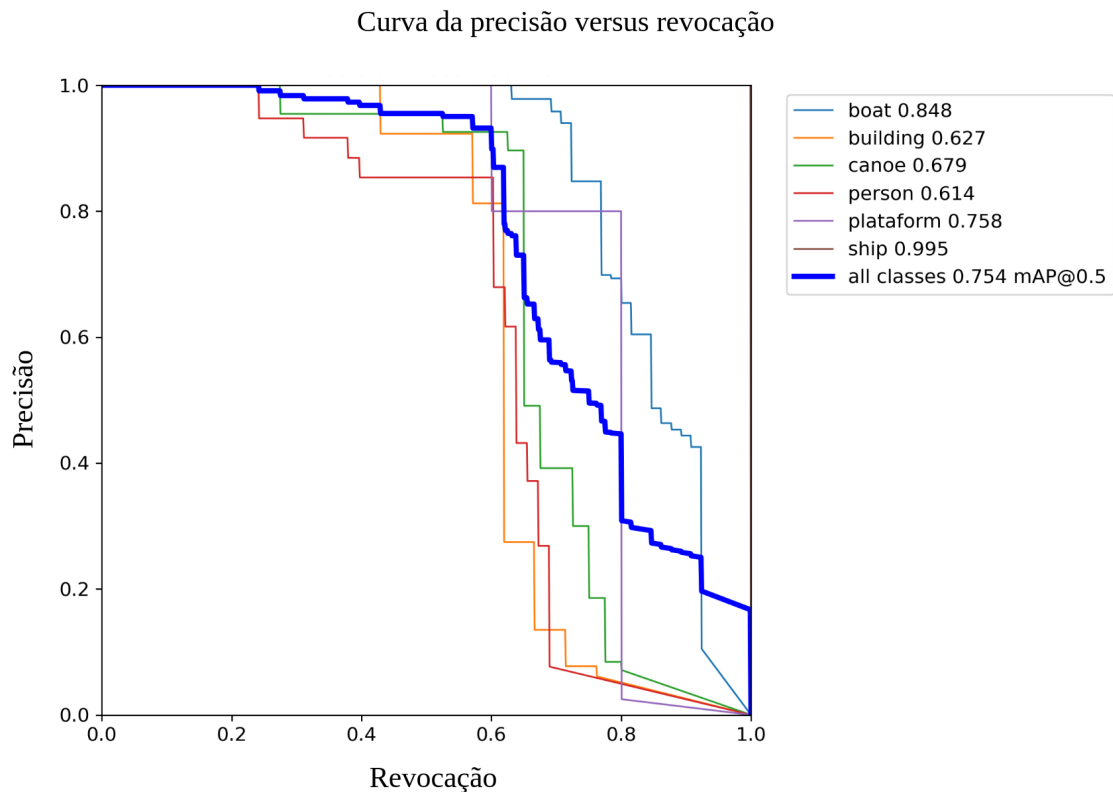
Por fim, por meio da análise das funções de perdas exibidas na Figura 28, observou-se que o modelo estava apresentando sinais de *overfitting*, isto é, apresentou dificuldades para detectar diferentes tipos de objetos (baixa capacidade de generalização). As causas que corroboram para esse cenário de *overfitting* podem estar relacionadas ao baixo volume de imagens disponíveis para o treinamento, bem como também baixo volume de objetos por categoria.

4.3.2 Análise da precisão e revocação

De forma semelhante à análise feita para o modelo treinado a partir da base de imagens *WSODD*, foram feitas também comparações em termos de precisão e revocação para o modelo treinado a partir de imagens regionais. Essas duas métricas, precisão e revocação, são essenciais para monitorar a performance do modelo de detecção de objetos. Figura 22-d e Figura 22-e ilustram como as métricas de precisão e revocação evoluíram ao longo do treinamento.

De maneira geral, ressalta-se que a precisão permite monitorar a taxa do número de acertos relativos aos falsos positivos. Falso positivo, no contexto de modelos de detecção de objetos, representa um objeto que pertence originalmente a uma classe qualquer, porém foi classificado pelo modelo como classe positiva (categoria de interesse). Por outro lado, a revocação permite monitorar o percentual de acertos relativos aos falsos negativos. Nesse contexto, falso negativo ocorre quando um objeto que pertence originalmente a classe positiva foi classificado como pertencente a classe negativa (categoria qualquer que não seja a de interesse). Considerando essas duas métricas, a análise delas permite entender se o modelo treinado está produzindo inferências incorretas mais em termos de falsos negativos ou mais em termos de falsos positivos. De modo geral, não é possível melhorar a precisão e revocação proporcionalmente e, por isso, a priorização pela redução de falsos negativos ou falsos positivos pode ser feita por meio da especificação do limiar (IoU) pelo qual o modelo deve se basear para inferir a classe de um objeto. A Figura 29, a seguir, ilustra o comportamento da curva de precisão versus curva de revocação.

Figura 29 – A figura ilustra a variação das métricas de precisão e revocação ao longo da escolha do par (precisão, revocação) desejado



Fonte: (Elaborado pelo autor, 2022).

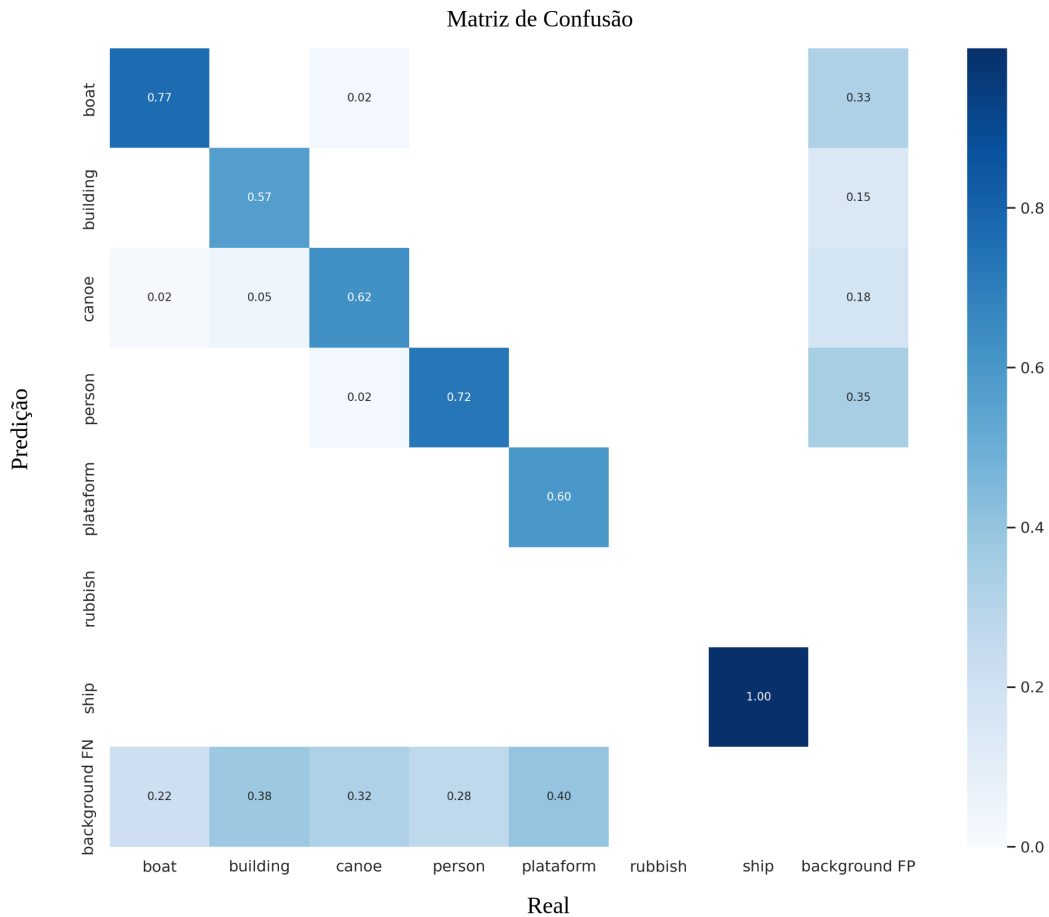
Por meio da figura acima, observa-se que o aumento da precisão implica em um custo da métrica de revocação e vice-versa. Em decorrência disso, especifica-se um limiar (IoU) onde seja obtida a máxima precisão pelo menor custo da revocação ou vice-versa. Em síntese, geralmente a escolha do limiar depende da necessidade da aplicação: priorizar a redução de falsos negativos ou priorizar a redução de falsos positivos. No caso da aplicação proposta neste trabalho, busca-se um limiar onde haja um equilíbrio entre as duas métricas. Sob essa ótica, foi escolhido o limiar (IoU) de 0.5 como base para a inferência das classes dos objetos. Ainda considerando a Figura 29, ao lado direito, observa-se a obtenção de uma precisão média (mAP) de 75.4 % obtida a partir do limiar de 0.5 especificado.

Por fim, o modelo de detecção de objetos treinado a partir de imagens regionais obteve uma performance razoável no que diz respeito as métricas de precisão e revocação. Observou-se também que é possível melhorá-las à medida que a base de imagens se torna mais volumosa e diversificada, especialmente para as classes *"building"*, *"canoe"* e *"person"* onde foram observadas as menores taxas de precisão.

A Figura 30 sintetiza o resultado das classificações para cada categoria por meio de uma matriz de confusão. Por meio dela, observa-se que as categorias que tiveram a

maior porcentagem de instâncias classificadas erroneamente foram as classes *building* e *plataform*. A razão disso pode ser em virtude da baixa quantidade de imagens e instância que representam essas classes no conjunto de treino.

Figura 30 – Matriz de confusão das classificações obtidas pelo modelo.

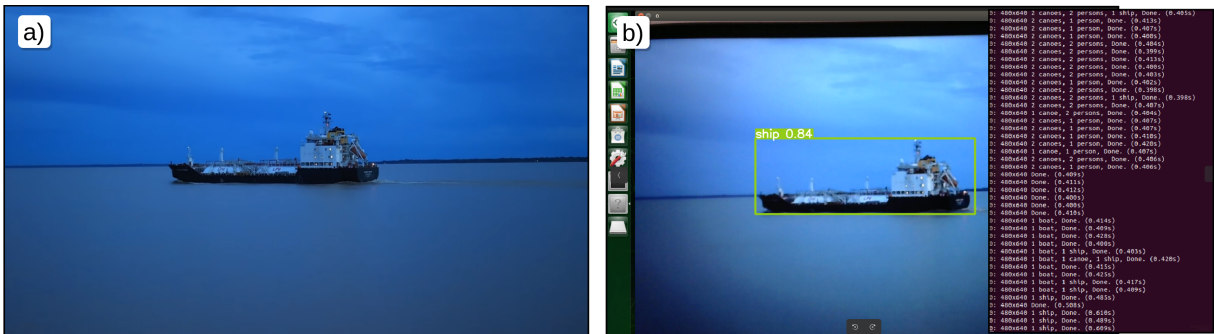


Fonte: (Elaborado pelo autor, 2022).

4.3.3 Análise de performance do modelo em uma plataforma embarcada

Conforme visto na Seção 3.3.7, a última etapa deste trabalho consistiu na integração do modelo treinado em um dispositivo de borda (plataforma embarcada *Jetson Nano*). A seguir, serão apresentadas análises que dizem respeito aos desempenho do modelo de detecção de objetos em função da taxa de quadros por segundo.

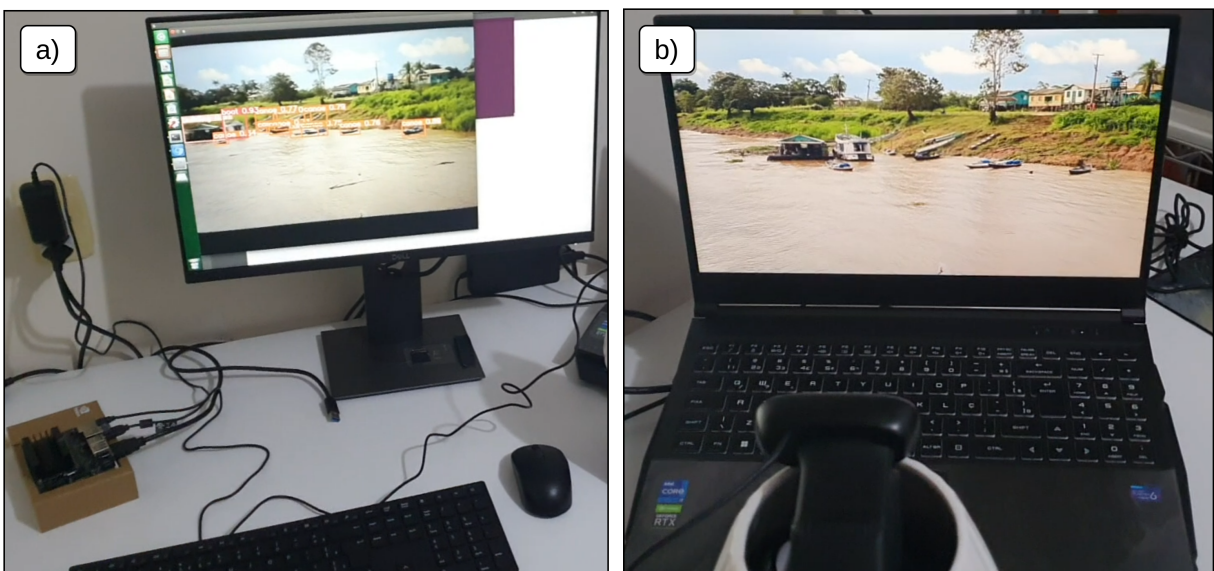
Figura 31 – Exemplo



Fonte: (Elaborado pelo autor, 2022).

A Figura 32 ilustra o procedimento adotado para simular um cenário real de detecção de objetos localizados em superfície de rios da Amazônia, bem como avaliar a performance do modelo em um dispositivo embarcado. Dessa forma, a Figura 32-b, à direita, ilustra o processo de aquisição de imagens a partir de um vídeo que retrata um cenário real de embarcações realizando o trajeto fluvial entre as cidades de Manaus e Tefé. Em seguida, conforme ilustra a Figura 32-a, os objetos são detectados em tempo real de execução.

Figura 32 – Resultados obtidos a partir da simulação de um cenário real de detecção de objetos com a plataforma embarcada *Jetson Nano*. À esquerda, a figura ilustra objetos sendo detectados à medida que a câmera, na figura ilustrada à direita, captura novas imagens.



Fonte: (Elaborado pelo autor, 2022).

A realização de testes em diferentes condições de iluminação, diferentes cenários e com objetos diversos foram fatores levados em consideração para os testes realizados com

a plataforma embarcada *Jetson Nano*. A Figura 33, por exemplo, exemplifica um trecho de um determinado vídeo onde uma embarcação é detectada em um período do dia com menor claridade.

Figura 33 – Ilustração de uma embarcação sendo detectada em um período com menor intensidade de raios luminosos.



Fonte: (Elaborado pelo autor, 2022).

De forma semelhante, a Figura 34 ilustra o processo de detecção de objetos sendo executado em um cenário com maior diversidade de objetos. Em média, a plataforma *Jetson Nano* conseguiu executar o modelo de detecção de objetos, treinado a partir do método YOLOv5, com uma taxa de 3 quadros por segundo, aproximadamente.

Figura 34 – Na figura é ilustrado uma detecção equívoca do modelo, onde ondas formadas pelo deslocamento do barco na superfície da água formaram um objeto com traços semelhantes a de uma canoa.



Fonte: (Elaborado pelo autor, 2022).

Em síntese, os resultados obtidos por meio deste trabalho indicam que é possível aperfeiçoar o modelo de detecção de objetos localizados em superfícies de rios regionais da Amazônia à medida que novas imagens são inseridas na base de dados criada, sendo possível também explorar novas técnicas de expansão de base de imagens e, ainda, testar modelos de detecção de objetos mais robustos.

5 CONCLUSÃO

Este trabalho apresentou avaliações de desempenho para detectores de objetos localizados em superfícies marítimas e em superfícies de rios da Amazônia. Para superfícies marítimas, foi utilizado a base de imagens WSODD (*Water Surface Object Detection Dataset*). Para superfícies de rios da Amazônia, foi criada e anotada a base de imagens WSOD-ARD (*Water Surface Object Detection - Amazon Rivers Dataset*). Ao longo do processo de criação da base de imagens, foram observados diversos desafios, tais como definição dos limites da caixa delimitadora para objetos oclusos, anotação de objetos com dimensões inferiores, baixo volume de imagens para categorias menos frequentes e entre outras dificuldades que enriqueceram o processo de maturação na criação de uma base de imagens anotadas com objetos comuns ao contexto de rios da região Amazônica. Para o detector treinado com a base de imagens WSODD, este obteve uma precisão média (mAP) equivalente à 76.3 %. Para o detector de objetos treinado com a base de imagens WSOD-ARD, foram utilizadas técnicas de Aprendizado por Transferência a fim de aproveitar os pesos do modelo treinado a partir da base de imagens WSODD, obtendo uma precisão média igual à 75.4 %. Por último, esse modelo foi integrado em um dispositivo de borda (plataforma embarcada Nvidia Jetson Nano) com o objetivo de simular um cenário real de utilização da aplicação proposta.

Por fim, trabalhos futuros poderão ser realizados tomando como ponto de partida os estudos comparativos realizado com o detector YOLOv5, apresentados neste trabalho. Entre os possíveis trabalhos futuros, citam-se: (i) avaliar e comparar modelos treinados a partir do algoritmo YOLOv7 (versão mais recente do detector de objetos YOLO até o momento da publicação deste trabalho); (ii) incrementar a base de imagens WSOD-ARD com imagens reais e capturadas em diferentes estações do ano, distintas condições de iluminação (ao amanhecer, ao entardecer, ao anoitecer) e distintas condições climáticas (dias ensolarados, nublados, nebulosos e chuvosos) de modo que seja possível criar uma base de imagens mais robusta; (iii) explorar técnicas em estado-da-arte que viabilizam o aumento da base de imagens. Dessa forma, com vistas a essas e outras possibilidades de futuras pesquisas, a base de imagens criada neste trabalho será disponibilizada publicamente para fins acadêmicos⁹.

Em decorrência da relevância do tema apresentado, uma versão reduzida deste trabalho foi escrita em formato de artigo¹⁰ e publicada nos Anais do XIV Simpósio Brasileiro de Computação Ubíqua e Pervasiva (2022), tendo sido também apresentado no Instituto de Computação de Niterói-RJ (UFF).

⁹O desenvolvimento deste trabalho, bem como acesso a base de imagens, pode ser realizada a partir desse repositório: <https://github.com/rTiagoS/water-surface-object-detection>

¹⁰Link para acessar o artigo publicado: <https://sol.sbc.org.br/index.php/sbcup/article/view/20608>

REFERÊNCIAS BIBLIOGRÁFICAS

- BLOISI, D. et al. Camera based target recognition for maritime awareness. In: **2012 15th International Conference on Information Fusion**. [S.l.: s.n.], 2012. p. 1982–1987.
- BLOISI, D. D. et al. Automatic maritime surveillance with visual target detection. In: . [S.l.: s.n.], 2011.
- DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2009. p. 248–255.
- EVERINGHAM, M. et al. The pascal visual object classes challenge: A retrospective. **International Journal of Computer Vision**, v. 111, n. 1, p. 98–136, jan. 2015.
- GIRSHICK, R. B. Fast R-CNN. **CoRR**, abs/1504.08083, 2015. Disponível em: <http://arxiv.org/abs/1504.08083>.
- GONZALEZ, R.; WOODS, R. **Processamento Digital De Imagens**. ADDISON WESLEY BRA, 2009. ISBN 9788576054016. Disponível em: <https://books.google.com.br/books?id=r5f0RgAACAAJ>.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.
- GU, Y. et al. Autonomous vessels: State of the art and potential opportunities in logistics. **SSRN Electronic Journal**, n. 2019/6, 01 2019.
- HE, K. et al. Mask R-CNN. **CoRR**, abs/1703.06870, 2017. Disponível em: <http://arxiv.org/abs/1703.06870>.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. In: **Intelligent Signal Processing**. [S.l.]: IEEE Press, 2001. p. 306–351.
- LI, X. et al. A modified yolov3 detection method for vision-based water surface garbage capture robot. **International Journal of Advanced Robotic Systems**, v. 17, n. 3, p. 1729881420932715, 2020. Disponível em: <https://doi.org/10.1177/1729881420932715>.
- LIN, T. et al. Focal loss for dense object detection. **CoRR**, abs/1708.02002, 2017. Disponível em: <http://arxiv.org/abs/1708.02002>.
- LIN, T. et al. Microsoft COCO: common objects in context. **CoRR**, abs/1405.0312, 2014. Disponível em: <http://arxiv.org/abs/1405.0312>.

- LIU, W. et al. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015. Disponível em: <http://arxiv.org/abs/1512.02325>.
- PRASAD, D. K. et al. Are object detection assessment criteria ready for maritime computer vision? *IEEE Transactions on Intelligent Transportation Systems*, v. 21, n. 12, p. 5295–5304, 2020.
- PRASAD, D. K. et al. Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. *IEEE Transactions on Intelligent Transportation Systems*, v. 18, n. 8, p. 1993–2016, 2017.
- REN, S. et al. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. Disponível em: <http://arxiv.org/abs/1506.01497>.
- RIBEIRO, R. et al. A data set for airborne maritime surveillance environments. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 29, n. 9, p. 2720–2732, 2019.
- WANG, W. et al. Roboat II: A novel autonomous surface vessel for urban environments. *CoRR*, abs/2007.10220, 2020. Disponível em: <https://arxiv.org/abs/2007.10220>.
- XU, Q. et al. Deep convolutional neural network-based autonomous marine vehicle maneuver. *International Journal of Fuzzy Systems*, v. 20, 09 2017.
- ZHAO, Z. et al. Object detection with deep learning: A review. *CoRR*, abs/1807.05511, 2018. Disponível em: <http://arxiv.org/abs/1807.05511>.
- ZHOU, B. et al. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 40, n. 6, p. 1452–1464, 2018.
- ZHOU, Z. et al. An image-based benchmark dataset and a novel object detector for water surface object detection. *Frontiers in Neurorobotics*, v. 15, 2021. ISSN 1662-5218. Disponível em: <https://www.frontiersin.org/article/10.3389/fnbot.2021.723336>.