

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
FACULDADE DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA  
ELÉTRICA

Fabício de Oliveira Coelho

Nova Metodologia para Exploração e Mapeamento Autônomo de  
VANTs em Terrenos Desconhecidos Usando *Deep Learning* Através de  
Estratégia Orientada por Fronteiras

Juiz de Fora

2025

**Fabício de Oliveira Coelho**

**Nova Metodologia para Exploração e Mapeamento Autônomo de  
VANTs em Terrenos Desconhecidos Usando *Deep Learning* Através de  
Estratégia Orientada por Fronteiras**

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Energia

Orientador: Prof. André Luís Marques Marcato, D.Sc.

Coorientadora: Prof. Milena Faria Pinto, D.Sc.

Juiz de Fora

2025

Ficha catalográfica elaborada através do programa de geração automática da Biblioteca Universitária da UFJF, com os dados fornecidos pelo(a) autor(a)

Coelho, Fabrício .

Nova Metodologia para Exploração e Mapeamento Autônomo de VANTs em Terrenos Desconhecidos Usando Deep Learning Através de Estratégia Orientada por Fronteiras / Fabrício Coelho. -- 2025.  
155 f. : il.

Orientador: André Luís Marcato

Coorientadora: Milena Pinto

Tese (doutorado) - Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Programa de Pós-Graduação em Engenharia Elétrica, 2025.

1. UAVs. 2. Deep Learning. 3. Robótica. 4. Missões Autônomas. 5. Inteligência Artificial. I. Marcato, André Luís, orient. II. Pinto, Milena, coorient. III. Título.

**Fabício de Oliveira Coelho**

**Nova Metodologia para Exploração e Mapeamento Autônomo de VANTs em Terrenos Desconhecidos Usando Deep Learning Através de Estratégia Orientada por Fronteiras**

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Sistemas de Energia Elétrica

Aprovada em 22 de maio de 2025.

**BANCA EXAMINADORA**

**Prof. Dr. André Luis Marques Marcato** - Orientador

Universidade Federal de Juiz de Fora

**Profa. Dra. Milena Faria Pinto** - Coorientadora

Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

**Prof. Dr. José Luís Sousa de Magalhães Lima**

Instituto Politécnico de Bragança

**Prof. Dr. Eduardo Todt**

Universidade Federal do Paraná

**Prof. Dr. Leonardo de Mello Honório**

Universidade Federal de Juiz de Fora

**Prof. Dr. Alexandre Bessa dos Santos**

Universidade Federal de Juiz de Fora

Juiz de Fora, 06/05/2025.



Documento assinado eletronicamente por **Andre Luis Marques Marcato, Professor(a)**, em 22/05/2025, às 14:53, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Milena Faria Pinto, Usuário Externo**, em 22/05/2025, às 15:57, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **José Luis Sousa Magalhães Lima, Usuário Externo**, em 23/05/2025, às 05:31, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Alexandre Bessa dos Santos, Professor(a)**, em 23/05/2025, às 07:37, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Eduardo Todt, Usuário Externo**, em 25/05/2025, às 16:07, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Leonardo de Mello Honorio, Professor(a)**, em 26/05/2025, às 09:54, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf ([www2.ufjf.br/SEI](http://www2.ufjf.br/SEI)) através do ícone Conferência de Documentos, informando o código verificador **2377560** e o código CRC **C55F6094**.

Dedico este trabalho, em primeiro lugar, a Deus,  
a minha família e amigos!

## AGRADECIMENTOS

Agradeço, primeiramente, a Deus por ter me dado todo o suporte espiritual necessário para que eu pudesse alcançar este momento em minha trajetória acadêmica. Sem Ele, nada disso seria possível.

Aos meus pais, expresso minha profunda gratidão por terem proporcionado as bases para minha educação e, hoje, serem um porto seguro em todas as decisões da minha vida. Estendo meu agradecimento a todos os meus familiares, em especial à minha irmã e aos meus avós, por toda a energia positiva e pelos momentos em família, que são verdadeiramente insubstituíveis.

Aos meus amigos, sou imensamente grato pelos momentos de alegria e descontração que sempre trouxeram leveza à caminhada.

Meu agradecimento também vai aos meus orientadores e professores, André Marcato e Milena Faria, que aceitaram a missão de me orientar desde os primeiros passos na graduação e seguiram comigo até o doutorado. Sem eles e tantos outros mestres excepcionais, este caminho teria sido muito mais difícil, talvez até inalcançável.

Por fim, expresso meu reconhecimento de gratidão à minha esposa, que esteve presente em todas as minhas dedicatórias acadêmicas anteriores e ocupa um papel essencial em minha vida. Tenho a imensa felicidade de tê-la ao meu lado, sendo não apenas minha companheira de vida, mas também uma fonte constante de inspiração, graças à sua excelência técnica e à sua habilidade singular na arte de ensinar. Agradeço de coração pelo apoio incondicional, pela compreensão e pela paciência, especialmente diante das demandas dos estudos, que tantas vezes exigiram ausências, mesmo estando fisicamente presente em casa.

"Se eu vi mais longe, foi por estar sobre ombros  
de gigantes."

Sir Isaac Newton.

## RESUMO

Os Veículos Aéreos Não Tripulados (VANTs) desempenham um papel fundamental na exploração de terrenos imprevisíveis, como locais de desastres e zonas de busca e resgate. Para operar com segurança e eficiência em ambientes dinâmicos, esses robôs exigem navegação autônoma precisa e planejamento de trajetória eficiente. Este trabalho apresenta uma estratégia inovadora para a exploração autônoma de VANTs em ambientes desconhecidos, com foco na detecção e utilização de fronteiras.

A pesquisa propõe métodos avançados baseados em técnicas de aprendizagem profunda para a detecção precisa de fronteiras, visando a expansão eficiente do mapa desconhecido no menor tempo possível. Além disso, introduz-se uma nova política de tomada de decisão que permite ao robô selecionar a fronteira mais adequada para exploração, utilizando informações derivadas de um modelo de rede neural especializado nessa tarefa.

Os resultados foram avaliados em três cenários distintos: ambiente simulado interno, ambiente simulado externo e um experimento prático em ambiente externo. A aplicação da metodologia proposta resultou em uma redução de aproximadamente 25% no tempo de exploração em ambientes internos e 51% em ambientes externos quando comparado à um trabalho recente considerado estado da arte. Além da redução do tempo, métricas como a distância média percorrida e o número de decisões tomadas também apresentaram melhorias em todos os cenários analisados.

Outro aspecto relevante da pesquisa foi a demonstração da fraca correlação entre o número de fronteiras detectadas e o tempo necessário para identificá-las, evidenciando a robustez da metodologia proposta. O resultado observado indica que o tempo de detecção de fronteiras permanece praticamente constante, independentemente da quantidade delas, tornando a abordagem especialmente adequada para ambientes desconhecidos e de alta complexidade.

Palavras-chave: Missões Autônomas. Exploração. Terreno Desconhecido. VANT.

## ABSTRACT

Unmanned Aerial Vehicles (UAVs) play a crucial role in exploring unpredictable terrains, such as disaster sites and search zones. To operate safely and efficiently in dynamic environments, these robots require precise autonomous navigation and efficient trajectory planning. This work presents an innovative strategy for the autonomous exploration of UAVs in unknown environments, focusing on boundary detection and utilization.

The research proposes advanced methods based on deep learning techniques for accurate boundary detection, aiming to expand the unknown map as efficiently as possible. Additionally, a new decision-making policy is introduced, enabling the robot to select the most suitable boundary for exploration using information derived from a neural network model specialized in this task.

The results were evaluated in three distinct scenarios: an indoor simulated environment, an outdoor simulated environment, and a real-world outdoor experiment. The proposed methodology reduced exploration time by approximately 25% indoors and 51% outdoors compared to a recent state-of-the-art approach. In addition to time efficiency, other metrics—such as average distance traveled and the number of decisions made—also improved across all scenarios.

Another relevant aspect of this research was the demonstration of a weak correlation between the number of detected boundaries and the time required to identify them, highlighting the robustness of the proposed methodology. The observed results indicate that boundary detection time remains nearly constant, regardless of the number of boundaries present, making this approach particularly suitable for unknown and highly complex environments.

Keywords: Autonomous Missions. Exploration. Unknown Terrain. UAV.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Drone RQ-900 Pertencente à FAB. . . . .	19
Figura 2 – Metodologia Proposta. . . . .	23
Figura 3 – Divisão em <i>Grid</i> de Uma Imagem. . . . .	40
Figura 4 – Estrutura de Saída com $B = 2$ e $N$ Classes. . . . .	42
Figura 5 – Estrutura da Rede Neural da YOLO. . . . .	43
Figura 6 – Evolução dos Algoritmos de Detecção de Objetos. . . . .	52
Figura 7 – Detecção das Caixas Delimitadoras. . . . .	54
Figura 8 – Estrutura Completa da YOLOv8. . . . .	60
Figura 9 – (a) Mapa 2D (b) Mapa 2.5D (c) Mapa 3D (d) Objeto Mapeado. . . . .	63
Figura 10 – Modelo Inicial da <i>Octree</i> . . . . .	65
Figura 11 – (a) Subdivisão em Novos 8 Octetos (b) Atualização dos Novos Estados dos Octetos. . . . .	66
Figura 12 – Expansão da Árvore Baseado no Método RRT: (a) Amostragem Aleatória. (b) Detecção do Nó Mais Próximo. (c) Projeção do Nó Mais Próximo até o Amostrado. (d) Incorporação do Nó Projetado à Árvore. . . . .	72
Figura 13 – Metodologia Informed-RRT*. . . . .	75
Figura 14 – Definição da Elipse no Informed-RRT*. . . . .	75
Figura 15 – Exemplo de Caminhos Gerados pelo Direct-DRRT*. . . . .	77
Figura 16 – Visão Geral da Metodologia. . . . .	79
Figura 17 – Exemplo de OctoMap Durante o Processo de Exploração. . . . .	80
Figura 18 – Exemplos de Imagens Geradas: (a) Nível 1 ( $z = 0.5m$ ) (b) Nível 7 ( $z = 3.5m$ ) (c) Nível 8 ( $z = 4.0m$ ) (d) Nível 9 ( $z = 4.5m$ ). . . . .	82
Figura 19 – Mapa Utilizado para Criar o <i>Dataset</i> . . . . .	83
Figura 20 – Exemplos de Fronteiras Para Anotação em Conjunto de Dados: (a) 9 Fronteiras (b) 6 Fronteiras. . . . .	84
Figura 21 – Avaliação Gráfica das Perdas Durante o Processo de Treinamento. . . . .	86
Figura 22 – Métricas de Desempenho Obtidas no Treinamento. . . . .	88
Figura 23 – (a) Arranjo de Fronteiras 1 (b) Arranjo de Fronteiras 2. . . . .	89

Figura 24 – Fronteiras Encontradas no: (a) Nível 1. (b) Nível 2. (c) Nível 3. (d) Nível 4. (e) Nível 5. (f) Nível 6. (g) Nível 7. (h) Nível 8.	91
Figura 25 – Ponto de Inflecção para Avaliação de $k$ .	93
Figura 26 – Metodologia Kneedle para Detecção do Ótimo $k$ .	94
Figura 27 – Clusterização: (a) As Fronteiras São Representadas Por Retângulos Vermelhos. (b) Clusters Oriundos do K-Means são Representados por Esferas Pretas.	94
Figura 28 – Problemas Observados ao se Utilizar a Distância Euclidiana como Custo de Deslocamento.	97
Figura 29 – Curvas Exponenciais Negativa e Positiva.	99
Figura 30 – Custo dos Clusters Quando $\lambda_R = \lambda_{DL}$ .	100
Figura 31 – Custo dos Clusters Quando $\lambda_R = \lambda_{DL} * 0.48988$ .	102
Figura 32 – Modelo de Drone Iris.	106
Figura 33 – Mapa <i>Indoor</i> a Ser Explorado.	108
Figura 34 – Exploração <i>Indoor</i> : (a) Representação do Mapa já Explorado com Fronteira. (b) Representação do Mapa já Explorado sem Fronteira.	109
Figura 35 – Comparação entre a metodologia Proposta $\times$ Batinovic et al. [28] em Ambientes <i>Indoor</i> .	110
Figura 36 – Caminho Percorrido no Mapa <i>Indoor</i> : (a) Metodologia Proposta (b) Metodologia de Batinovic et al. [28].	112
Figura 37 – Análise de Eficiência Computacional na Detecção de Fronteiras.	113
Figura 38 – Análise Comparativa da Frequência de Tomada de Decisão.	114
Figura 39 – Análise Comparativa da Eficiência de Exploração.	115
Figura 40 – Mapa <i>Outdoor</i> a Ser Explorado.	116
Figura 41 – Comparação Entre a Metodologia Proposta $\times$ Batinovic et al. [28] em Ambientes <i>Outdoor</i> .	118
Figura 42 – Caminho Percorrido no Mapa <i>Outdoor</i> : (a) Metodologia Proposta (b) Metodologia de Batinovic et al. [28].	119
Figura 43 – Comparação Entre Metodologias com Base na Variação de Altitude Durante a Exploração.	120
Figura 44 – Análise Comparativa da Frequência de Tomada de Decisão.	121

Figura 45 – Análise de Eficiência Computacional na Detecção de Fronteiras. . . . .	122
Figura 46 – Tarefa de Inspeccionamento via Contato Físico pelo Drone do Projeto AeroARMs. . . . .	124
Figura 47 – Caminho Percorrido no Ambiente Real: (a) Missão 1 (b) Missão 2. . . . .	126
Figura 48 – Análise de Eficiência Computacional na Detecção de Fronteiras para a Missão 1. . . . .	126
Figura 49 – Análise de Eficiência Computacional na Detecção de Fronteiras para a Missão 2. . . . .	127
Figura 50 – (a) Cenário 1 - Com Fronteiras (b) Cenário 1 - Sem Fronteiras.	130

## LISTA DE TABELAS

Tabela 1 – Metodologias de Detecção de Objetos [51]. . . . .	50
Tabela 2 – Análise Quantitativa de Missões de Exploração Autônoma.	111
Tabela 3 – Distância Total Percorrida pelo Drone ( $m$ ). . . . .	112
Tabela 4 – Análise Quantitativa de Missões de Exploração Autônoma.	117
Tabela 5 – Distância Total Percorrida Pelo Drone ( $m$ ). . . . .	118
Tabela 6 – Análise Quantitativa da Variação de Altura em Navegação <i>Out-door</i> . . . . .	120
Tabela 7 – Análise Comparativa da Detecção de Fronteiras em uma Imagem. . . . .	129
Tabela 8 – Tempo Necessário para Detectar Todas as Fronteiras (ms).	130
Tabela 9 – Correlação Entre os Tempos de Detecção de Fronteiras e a Quantidade de Fronteiras. . . . .	134
Tabela 10 – Classificação de Correlações Segundo Evans [126]. . . . .	134
Tabela 11 – Tempos Médios para a Detecção de Fronteiras. . . . .	135

## LISTA DE ABREVIATURAS E SIGLAS

C2F	<i>Coarse-to-Fine</i>
CNN	<i>Convolutional Neural Network</i>
CPP	<i>Coverage Path Planning</i>
CPU	<i>Central Processing Unit</i>
DFL	<i>Distributed Focal Loss</i>
DPM	<i>Deformable Parts Models</i>
FAB	Força Aérea Brasileira
GAN	<i>Generative Adversarial Network</i>
FAEP	<i>Fast Autonomous Exploration Planner</i>
FIS	<i>Frontier Information Structure</i>
FoV	<i>Field of View</i>
FSOTSP	<i>Fixed Start Open Travelling Salesman Problem</i>
FUEL	<i>Fast UAV Exploration</i>
GPS	<i>Global Positioning System</i>
HOG	<i>Histograms of Oriented Gradient</i>
IMU	<i>Inertial Measurement Unit</i>
IA	<i>Inteligência Artificial</i>
IoT	<i>Internet of Things</i>
IoU	<i>Intersection Over Union</i>
LFE	<i>Lightweight Frontier Exploration</i>
LiDAR	<i>Light Detection and Ranging</i>
mAP	<i>Mean Average Precision</i>
PRM	<i>Probabilistic Road Maps</i>
RTT	<i>Rapidly-exploring Random Trees</i>
ROS	<i>Robot Operating System</i>
SAR	<i>Search and Rescue</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
UAV	<i>Unmanned Aerial Vehicle</i>
VANT	Veículo Aéreo Não Tripulado
YOLO	<i>You Only Look Once</i>
WFD	<i>Wavefront Frontier Detector</i>
SSE	<i>Sum of Squared Error</i>
MAP	<i>Mean Average Precision</i>

NBV	<i>Next-Best-View</i>
PCATSP	<i>Priority Constrained Asymmetric Traveling Salesman Problem</i>
ReLU	<i>rectified Linear Unit</i>
ROI	<i>Region of Interesting</i>
RPN	<i>region proposal network</i>
SGD	<i>Stochastic Gradient Descent</i>
SiLU	<i>Sigmoid Linear Unit</i>
SPPF	<i>Spatial Pyramid Pooling Fast</i>
SSE	<i>sum of squared errors</i>
SVM	<i>Support Vector Machine</i>
4IR	<i>Quarta Revolução Industrial</i>
IMU	<i>Inertial Measurement Units</i>
EKF	<i>Extended Kalman Filter</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
1.1	METODOLOGIA PROPOSTA E SUAS PRINCIPAIS CONTRIBUIÇÕES	20
1.2	ORGANIZAÇÃO	23
1.3	CONTRIBUIÇÕES	24
<b>2</b>	<b>TRABALHOS RELACIONADOS</b>	<b>26</b>
2.1	METODOLOGIA PROPOSTA × VSLAM	33
<b>3</b>	<b>FUNDAMENTOS TEÓRICOS</b>	<b>36</b>
3.1	METODOLOGIA PARA DETECÇÃO DE OBJETOS - YOLO	36
3.1.1	Introdução à YOLO	36
3.1.2	Modelagem YOLO	40
3.1.3	Arquitetura de Rede YOLO	42
3.1.4	Treinamento da Rede YOLO	45
3.1.5	Versões Baseadas na YOLO	51
3.1.6	Detecção de Objetos - YOLOv8	59
3.2	MAPAS 3D UTILIZANDO A ESTRUTURA OCTOMAP	62
3.3	PLANEJAMENTO DE CAMINHOS	70
3.3.1	Variante Informed-RRT*	74
3.3.2	Variante Direct-DRRT*	76
<b>4</b>	<b>METODOLOGIA PROPOSTA</b>	<b>78</b>
4.1	EXTRAÇÃO DE DADOS DO AMBIENTE	78
4.2	SÍNTESE DO MODELO <i>DEEP LEARNING</i> PARA DETECÇÃO DE FRONTEIRAS	81
4.2.1	Análise do Treinamento da Rede	85
4.3	CLUSTERIZAÇÃO DE FRONTEIRAS	89
4.4	TOMADA DE DECISÃO NA SELEÇÃO DA MELHOR FRONTEIRA	95
4.5	ALGORITMO DE EXPLORAÇÃO AUTÔNOMA BASEADO EM FRONTEIRAS UTILIZANDO <i>DEEP LEARNING</i>	102
<b>5</b>	<b>RESULTADOS FINAIS</b>	<b>105</b>
5.1	EXPLORAÇÃO AUTÔNOMA EM AMBIENTES <i>INDOOR</i>	108

5.2	EXPLORAÇÃO AUTÔNOMA EM AMBIENTES <i>OUTDOOR</i> .	116
5.3	DETECÇÃO DE FRONTEIRAS EM UM AMBIENTE REAL .	123
5.4	ANÁLISE DO DESEMPENHO PARA A EXTRAÇÃO DE FRONTEIRAS . . . . .	128
5.4.1	Análise Estatística da Correlação Entre o Tempo de Detecção de Fronteiras e o Volume Conhecido . . . . .	131
<b>6</b>	<b>CONCLUSÕES . . . . .</b>	<b>136</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>140</b>

## 1 INTRODUÇÃO

Na era da Quarta Revolução Industrial (4IR), os Veículos Aéreos Não Tripulados (VANTs) estão pavimentando o caminho para soluções inovadoras em tecnologia avançada, promovendo abordagens eficientes e sustentáveis em uma variedade de campos de pesquisa e aplicação [1]. Esses avanços permeiam diversos domínios, incluindo Agricultura 4.0 [2], Indústria 4.0 [3], logística [4], inteligência artificial [5], Internet das Coisas (IoT) [6], sensoriamento remoto [7], e muitos outros.

Nas últimas décadas, os VANTs emergiram como ferramentas potenciais para explorar terrenos [8, 9, 10]. Esses terrenos, frequentemente incertos, podem incluir locais de acidentes, zonas de busca e áreas de reconhecimento [11, 12]. Em tais cenários, o planejamento da missão desempenha um papel crucial, permitindo que os VANTs localizem o alvo durante a missão enquanto navegam por um caminho global predefinido [13]. Essa convergência de tecnologia avançada e planejamento preciso da missão aprimora as capacidades dos VANTs para explorar terrenos imprevisíveis e fortalece suas aplicações potenciais em diferentes setores, impulsionando ainda mais a era 4IR.

Em ambientes dinâmicos, os potenciais obstáculos representam desafios significativos para os VANTs. Nestes casos, é essencial que os VANTs possuam autonomia para executar diversas operações e tomar decisões com base nas informações disponíveis. O planejamento da missão requer acesso aos dados do ambiente para assegurar confiabilidade, segurança e eficiência durante as operações [14]. Uma etapa fundamental no planejamento da missão é o mapeamento preciso, pois desempenha um papel crítico nas técnicas bem-sucedidas de planejamento de trajetória, que dependem de dados detalhados do ambiente para identificar obstáculos ao longo do processo de definição de trajetória [15]. Para mitigar problemas e contornar obstáculos, é essencial considerar restrições como precisão de navegação, manobrabilidade, tempo de chegada e limitações de vida útil da bateria, garantindo assim uma operação segura e eficiente.

A necessidade da atuação desses tipos de robôs em ambientes desconhecidos pode ser observada por meio de notícias recentes sobre sua aplicação em tarefas

de resgate após um terremoto ocorrido na Turquia em 2023 [16]. Após o evento, VANTs equipados com sensores para mapeamento do novo ambiente, devido à destruição causada, realizaram missões sucessivas para reconstruir um mapa 3D. Com base nesse novo mapa, foi possível coordenar esforços com as autoridades locais para traçar a melhor estratégia para os resgates.

Outra opção para o uso de drones em ambientes de desastre é o auxílio ao corpo de bombeiros na extinção de incêndios, como mostrado em uma reportagem e entrevista com o bombeiro John Walkie, que utiliza esses robôs há 6 anos em seu trabalho [17]. O entrevistado explicou a importância do uso de drones para o reconhecimento do cenário, a fim de evitar possíveis áreas perigosas onde acidentes com colegas de trabalho poderiam ocorrer.

No trabalho de Wankmuller et al. [18], é realizada uma análise da aplicabilidade de drones em ambientes de desastres, identificando possíveis áreas de interesse para essa aplicação de salvamento, tais como: resgate em cavernas, operações de resgate em montanhas, entrega de suprimentos médicos e busca e resgate em áreas afetadas por avalanches. Além disso, o estudo apresenta algumas justificativas que tornam o uso de VANTs fundamental para esse tipo de aplicação, incluindo o baixo custo de implementação, tornando-os acessíveis para equipes de salvamento.

No trabalho apresentado por Pavkovic et al. [19], foi realizada uma discussão sobre o gerenciamento de emergências, dividido em quatro fases: preparação, resposta, resgate e resolução. O uso de drones para auxiliar nesse tipo de tarefa ocorre principalmente na primeira fase, de preparação. Nessa etapa, é possível coletar informações relevantes do ambiente, possibilitando uma resposta eficaz para neutralizar os perigos apresentados pelo cenário de desastre. Todas essas atividades destacam a necessidade comum de exploração e reconhecimento do ambiente.

Em 2024, a região sul do Brasil foi severamente afetada por enchentes que inundaram cidades inteiras e apagaram regiões antes habitadas do mapa. Durante dias, muitas pessoas ficaram isoladas, com acesso possível apenas por embarcações e aeronaves. Para auxiliar nas operações de busca e resgate, a Força Aérea Brasileira (FAB) utilizou VANTs. O drone utilizado, modelo RQ-900 de origem israelense (Figura 1), operado remotamente, não possuía capacidade para resgatar pessoas diretamente, mas estava equipado com câmeras de alta resolução capazes de

identificar pessoas no solo. Este VANT explorou uma área de aproximadamente 1.351 km<sup>2</sup> durante os esforços de busca, ajudando a localizar indivíduos ilhados e facilitar as operações de resgate realizadas por outros meios [20].

Figura 1 – Drone RQ-900 Pertencente à FAB.



Fonte: Retirado de [20]

Os exemplos analisados demonstram que os drones constituem ferramentas valiosas para operações em locais inacessíveis ou de alto risco para seres humanos. Contudo, sua operação eficaz requer profissionais qualificados, capazes de tomar decisões críticas durante as missões. Essas decisões tornam-se particularmente complexas devido a dois fatores principais: (i) o grande volume de dados gerados pelos sensores embarcados e (ii) o potencial impacto emocional envolvido em situações de risco. Um aspecto comum a todos os cenários estudados é a dependência de operações manuais para o controle e condução das missões.

Diante dessas dificuldades e dependências, a automatização das tarefas de exploração é fundamental. Isso permitirá que os drones operem em ambientes perigosos sem a interferência dos fatores de risco mencionados. Este trabalho aborda missões de exploração em ambientes desconhecidos, onde o drone deve mapear o maior território possível no menor tempo, garantindo a autonomia da missão. Como definido por Yamauchi em 1997, uma boa missão autônoma é aquela que explora a maior quantidade de áreas do ambiente dentro de um tempo razoável ou viável [21]. Portanto, toda a metodologia desenvolvida neste trabalho será direcionada a alcançar esse objetivo central.

Na exploração, é fulcral que o VANT se dirija exclusivamente a áreas não

mapeadas. Portanto, direcionar o drone para locais já explorados é ineficaz e deve ser evitado. A incorporação de inteligência aos drones para operação autônoma tem sido o foco de inúmeras pesquisas recentes. Análises de dados de sensores e tomadas de decisão, considerando o custo da exploração, são significativamente mais eficientes quando realizadas por sistemas computacionais especializados para essa finalidade. Esses sistemas são capazes de processar grandes volumes de dados em tempo real, permitindo que o VANT adapte sua missão de forma dinâmica e eficiente, evitando redundâncias e otimizando a cobertura de áreas desconhecidas.

## 1.1 METODOLOGIA PROPOSTA E SUAS PRINCIPAIS CONTRIBUIÇÕES

A presente pesquisa aborda os desafios relacionados à navegação autônoma em terrenos desconhecidos e potencialmente perigosos. O objetivo primordial é aprimorar as capacidades dos VANTs no enfrentamento de desafios críticos relacionados à exploração e mapeamento de áreas desconhecidas. Incluindo locais de acidentes, áreas que demandam missões de busca e reconhecimento, bem como locais nos quais o acesso humano deve ser restrito até que um conhecimento abrangente do local seja alcançado.

Nesse contexto, destaca-se o desenvolvimento de uma estratégia de mapeamento inteligente e eficiente, com ênfase na otimização da autonomia de voo dos VANTs. Essa abordagem considera as restrições energéticas impostas pela capacidade das baterias, o que exige a execução de missões de exploração no menor tempo possível sem comprometer a qualidade dos dados adquiridos.

O problema a ser resolvido de exploração autônoma pode ser subdividido em cinco questões cruciais em tarefas de robôs autônomos: localização, mapeamento, detecção de fronteiras, tomada de decisão e planejamento de caminhos. Durante todo o período de exploração, o robô deve manter uma localização precisa para garantir que o mapa em construção seja o mais próximo da realidade possível. Em ambientes de desastre, os obstáculos são considerados dinâmicos, portanto, são necessárias metodologias sensíveis às mudanças no ambiente. A partir do mapa, as fronteiras que representam as posições candidatas para expansão são identificadas. Em seguida, é necessário escolher entre as áreas existentes aquela que proporcionará o maior benefício em termos de expansão do ambiente. Essa etapa é denominada

tomada de decisão e sua abordagem está diretamente relacionada à eficiência da navegação. Por fim, com a posição objetiva definida, um planejador de caminho é executado para guiar o VANT com segurança da posição atual até a melhor posição candidata.

As tarefas de detecção de fronteiras e tomadas de decisão são fundamentais na exploração, pois constituem os pilares que irão guiar o robô para o objetivo final. A abordagem exploratória é definida pela execução sequencial dessas duas tarefas. O foco principal deste trabalho é apresentar uma nova metodologia capaz de detectar fronteiras e tomar decisões inteligentes. No entanto, para garantir a autonomia da missão, também serão consideradas as tarefas de mapeamento e planejamento de caminhos. Essas etapas são essenciais para permitir que o robô se desloque pelo ambiente de forma autônoma.

A metodologia adotada neste estudo visa direcionar o robô para áreas ainda desconhecidas. O acesso a essas regiões é realizado através das fronteiras, entendidas como a demarcação entre uma zona desconhecida e outra já explorada, livre de obstáculos cuja a definição foi apresentada em [21]. Essa definição revela-se crucial para a execução eficiente da exploração, assegurando que, ao ser identificada como uma fronteira, o robô seja capaz de alcançar essa posição sem impedimentos. Esta premissa estratégica proporciona um método eficaz para orientar o robô na exploração de territórios inexplorados.

A exploração autônoma neste trabalho começa com a tarefa de mapeamento, realizada por meio da técnica OctoMap [22, 23]. Nessa abordagem, os sensores do VANT constroem um mapa tridimensional detalhado do ambiente explorado por meio da leitura de sensores 3D. A estrutura resultante do OctoMap é então convertida em imagens para a aplicação de técnicas de aprendizagem profundo (do inglês, *deep learning*) voltadas para a localização de fronteiras. A metodologia escolhida para analisar essas imagens é uma rede neural, capaz de realizar inferências em tempo real de modelos previamente treinados.

A partir das numerosas fronteiras identificadas, é conduzido um processo de clusterização para agrupar de forma inteligente as fronteiras vizinhas, utilizando a técnica K-Means. Após a organização das fronteiras em clusters, uma função é aplicada para ponderar os custos associados à execução da missão em cada

cluster, determinando qual deles o robô deve priorizar. Finalmente, é empregada a metodologia Informed-RRT\* (*Rapidly-exploring Random Tree*) [24] para gerar um caminho livre de obstáculos, permitindo assim que o robô se mova de forma eficiente.

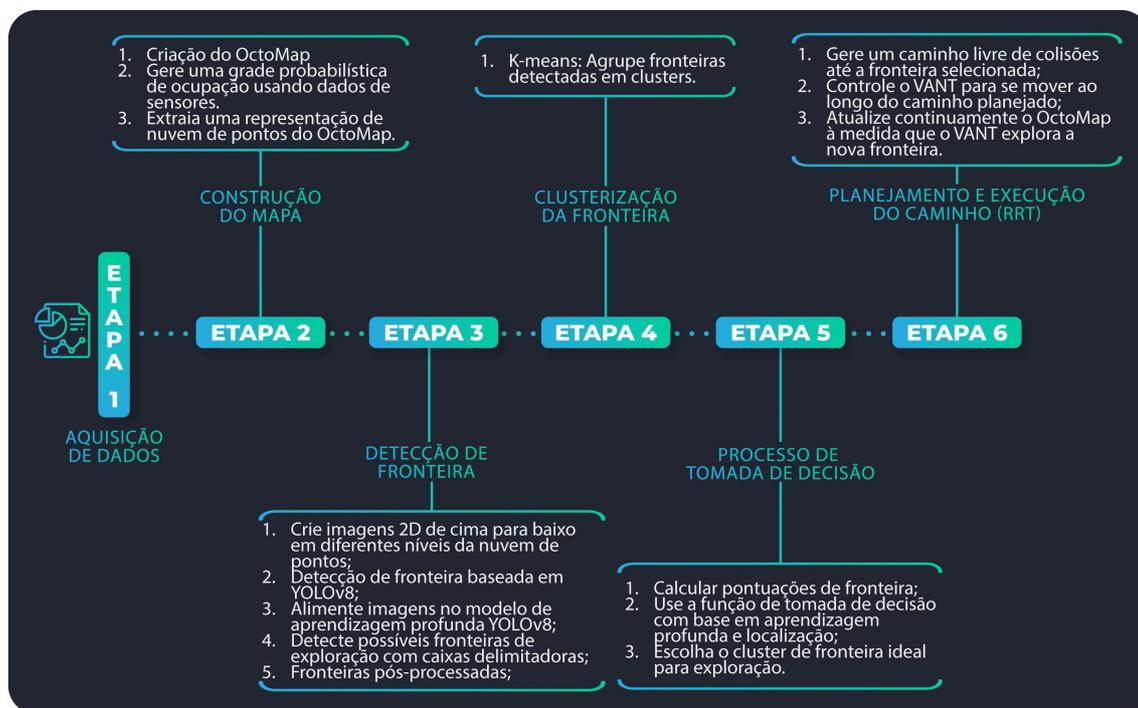
Diante da metodologia apresentada, as principais contribuições deste trabalho podem ser destacadas da seguinte maneira:

- **Proposição de uma abordagem inovadora para a otimização da expansão e refinamento de mapas em terrenos desconhecidos**, com foco em aprimorar a eficiência operacional de VANTs por meio de inteligência artificial. A implementação de missões autônomas baseadas em *deep learning* para a detecção de fronteiras surge como uma alternativa às metodologias tradicionais, nas quais o tempo necessário para identificar fronteiras está diretamente relacionado à quantidade de fronteiras no ambiente. Em cenários onde a quantidade de fronteiras é desconhecida, essa característica pode ser determinante para a eficiência computacional. A adoção de *deep learning* agrega um caráter determinístico à resolução do problema de exploração autônoma baseada em fronteiras, apresentando uma contribuição inédita e ainda não explorada na literatura existente;
- **Introdução de uma função para a seleção otimizada de fronteiras**, orientando o deslocamento do robô em direção a essas regiões estratégicas. Essa função integrará informações provenientes da rede neural responsável pela detecção de fronteiras e se diferenciará das abordagens convencionais ao empregar um planejador de caminhos rápido e eficiente para calcular os custos de deslocamento entre a posição atual do robô e os candidatos à melhor fronteira;
- **Avaliação da eficiência computacional da metodologia proposta**, com foco nos tempos necessários para a detecção de fronteiras e a execução das missões. Essa análise permitirá evidenciar a robustez e a viabilidade prática da abordagem desenvolvida, destacando seus benefícios em cenários reais de aplicação.

Todas essas contribuições serão destacadas como pontos de extrema importância na metodologia para a exploração autônoma de VANTs em ambientes totalmente desconhecidos. Esses elementos representam avanços significativos, fundamentais para o sucesso da operação autônoma dos VANTs em terrenos desafiadores e inexplorados.

Os elementos que compõem a metodologia proposta, bem como a forma como interagem entre si, podem ser visualizados na Figura 2.

Figura 2 – Metodologia Proposta.



Fonte: Elaborado pelo Autor.

## 1.2 ORGANIZAÇÃO

Este trabalho está estruturado em sete capítulos. O primeiro capítulo, de caráter introdutório, apresenta a motivação do estudo com base na literatura existente e aborda pesquisas que típicam o problema a ser explorado. No capítulo 2,

é realizada uma fundamentação teórica por meio de uma revisão bibliográfica das literaturas atuais, cujos objetivos se alinham com o propósito deste trabalho.

O capítulo 3 oferece a fundamentação teórica das ferramentas utilizadas para a implementação da exploração autônoma em drones. Em seguida, no capítulo 4, é detalhada a integração de todas as ferramentas mencionadas no capítulo 3 para conceber a metodologia proposta neste trabalho.

Os resultados parciais observados até o momento são apresentados no capítulo 5, enquanto as conclusões derivadas desses resultados são discutidas no capítulo 6.

### 1.3 CONTRIBUIÇÕES

Até o presente momento, esta tese de doutorado resultou em um artigo de revista:

- de Coelho, Fabrício Oliveira, et al. "Autonomous UAV Exploration and Mapping in Uncharted Terrain through Boundary-Driven Strategy." *IEEE Access* (2024).

Como contribuições indiretas para a concepção deste trabalho, destacam-se dois artigos desenvolvidos pelo autor desta tese de doutorado, cujo foco principal foi o desenvolvimento de estruturas de redes neurais para aplicações em sinais biológicos.

- Coelho, Fabrício, et al. "sEMG Signals Classification using CNN Features Extraction as a Reliable Method". *Congresso Brasileiro de Automática-CBA*. Vol. 2. No. 1. (2020).
- Coelho, Fabrício, et al. "A novel sEMG data augmentation based on WGAN-GP". *Computer methods in biomechanics and biomedical engineering* 26.9 (2023).

O primeiro artigo visa apresentar uma nova estrutura de rede neural voltada para a extração de características, com o objetivo de classificar sinais de

eletromiografia. O segundo artigo, publicado em um periódico, concentrou-se na criação de dados sintéticos por meio de redes adversariais generativas (GANs) para expandir um banco de dados. Esses trabalhos contribuíram significativamente para a consolidação dos conceitos e conhecimentos em *deep learning*, os quais foram aplicados de maneira eficaz neste estudo.

Como coautor de trabalhos acadêmicos, destaco três estudos correlacionados ao uso de VANTs realizados durante o curso de doutorado.

- Melo, Aurelio G., et al. "Dynamic optimization and heuristics based online coverage path planning in 3D environment for UAVs". *Sensors* 21.4 (2021).
- Biundini, Iago Z., et al. "Experimentation and simulation with autonomous coverage path planning for UAVs". *Journal of Intelligent & Robotic Systems* 105.2 (2022).
- Ramos, Gabryel S., et al. "Hybrid methodology based on computational vision and sensor fusion for assisting autonomous UAV on offshore messenger cable transfer operation". *Robotica* 40.8 (2022).

## 2 TRABALHOS RELACIONADOS

A exploração de ambientes desconhecidos utilizando VANTs tem despertado considerável interesse nos últimos anos na comunidade acadêmica [25]. Desta forma, esta seção revisa de forma abrangente as metodologias e técnicas de ponta empregadas nesse domínio. É importante ressaltar que, conforme definido na introdução deste trabalho, a resolução do problema de exploração envolve a subdivisão em cinco tarefas que o robô deve desempenhar, que serão detalhadas ainda nessa seção. A revisão bibliográfica deste trabalho priorizou a apresentação de estudos que abordam a solução completa ou a maior parte dos problemas relacionados à exploração autônoma. No entanto, será dedicada uma atenção especial à tarefa de detecção de fronteiras, uma vez que a metodologia proposta neste trabalho apresenta o maior teor de inovação na obtenção das fronteiras para esta tarefa.

Exploração autônoma é um tema central na pesquisa robótica, com uma vasta gama de abordagens documentadas na literatura. Muitos desses estudos priorizam decisões que se baseiam exclusivamente na situação imediata do robô, ou adotam uma estratégia que seleciona a fronteira mais próxima (conhecida como abordagem "ambiciosa"). No entanto, tais abordagens tendem a desconsiderar informações globais do ambiente, limitando a eficiência da exploração. Em contraste, uma estratégia mais informada, que leve em conta o contexto geral do ambiente, pode resultar em uma exploração mais eficiente [26].

Outro desafio enfrentado na exploração autônoma é a complexidade dos algoritmos de busca por fronteiras, que muitas vezes são computacionalmente intensivos e dependem do número de fronteiras presentes no ambiente. Essa complexidade pode impactar a capacidade do sistema em fornecer respostas rápidas e em tempo hábil durante a exploração [27].

Essas considerações destacam a importância de uma abordagem cuidadosa na concepção de algoritmos para exploração autônoma, visando tanto a eficiência computacional quanto a capacidade de tomar decisões informadas com base no contexto global do ambiente.

O tema da presente pesquisa é relevante não apenas para o VANT, o qual foi utilizado neste trabalho, mas também para robôs móveis, como demonstrado

em um dos estudos pioneiros na busca por fronteiras [21]. Embora a metodologia desse estudo seja originalmente direcionada a robôs móveis, ela também pode ser aplicada em contextos envolvendo VANTs [28]. Essa metodologia, simples em sua concepção, consiste em buscar regiões desconhecidas em um mapa de grade de ocupação (do inglês, *occupancy grid*) que toquem áreas conhecidas, definindo, assim, as fronteiras. No entanto, é importante destacar que essa abordagem é computacionalmente intensiva, o que pode aumentar significativamente os desafios, especialmente em ambientes tridimensionais. Apesar disso, essa metodologia ainda é amplamente reconhecida como a abordagem clássica para a obtenção de fronteiras. Na publicação referenciada, a tomada de decisão para a escolha da melhor fronteira para navegação é baseada na fronteira mais próxima do robô (abordagem ambiciosa). Essa estratégia simplifica o processo de decisão, entretanto fronteiras mais distantes com potencial de ganho de informação maiores serão ignoradas.

A metodologia de exploração autônoma de ambientes, denominada FUEL (do inglês, *Fast UAV ExpLoration*), introduziu uma nova abordagem para detecção de fronteiras, denominada *Frontier Information Structure* (FIS) [26]. Os autores propuseram uma estrutura que inclui informações específicas sobre os *voxels* caracterizados como fronteiras, os clusters que representam esses *voxels* e suas posições médias. Além disso, é contabilizado um valor de custo que corresponde ao custo de todas as conexões de fronteiras que formam os clusters. Em cada atualização do mapa, o algoritmo proposto detecta as fronteiras que deixaram de existir e os novos *voxels* que se tornam candidatos a fronteiras. Essa interação remete à abordagem clássica de busca por fronteiras, cuja complexidade aumenta com a quantidade de fronteiras processadas. Para contornar essa situação, os autores propuseram uma abordagem incremental, que basicamente analisa as novas fronteiras de acordo com a porção da área observada. Essa melhoria é interessante, pois acelera o processo, eliminando a necessidade de verificar a inclusão e remoção de fronteiras que não foram observadas na atualização. No entanto, essa abordagem não elimina a dependência do número de fronteiras observadas.

A metodologia empregada pela FUEL para orientar o robô na escolha de qual fronteira ou cluster seguir é baseada em um processo de amostragem uniforme em torno de um cilindro que envolve o cluster, conforme pode ser detectado

pelo sensor laser presente no robô. Cada cluster é avaliado através da soma das amostras que ocupam uma região livre de obstáculos. Essas pontuações são então ordenadas de forma decrescente, permitindo que o algoritmo estabeleça um plano global dos clusters a serem explorados. Essa abordagem proporciona uma maneira sistemática e eficiente de decidir os próximos passos do robô durante a exploração autônoma. Uma limitação significativa do sistema de ranqueamento de clusters reside na ausência de ponderação pelo tamanho das fronteiras geradoras. Como os parâmetros geométricos dos cilindros são fixos, o método atribui pontuações idênticas a fronteiras de dimensões radicalmente distintas - um claro falseamento da representatividade espacial. Essa isonomia artificial desconsidera a relevância física e compromete a discriminação espacial. Em ambientes dimensionalmente heterogêneos, essa abordagem gera distorções duplas: zonas operacionais relevantes são subestimadas, enquanto elementos menores são indevidamente destacados, invalidando a representação espacial fidedigna [26].

A metodologia FUEL revelou resultados promissores, tendo uma segunda versão denominada FAEP (do inglês, *Fast Autonomous Exploration Planner*) [27]. Os autores identificaram o fenômeno do *stop-and-go* caracterizando o comportamento do drone, que frequentemente pausa para ajustes de rota durante a exploração. Isso ocorre devido à ocorrência frequente de novas fronteiras durante a missão de exploração, que podem ser mais vantajosas do que as já existentes. Conseqüentemente, o algoritmo de planejamento global precisa ser constantemente adaptado. Como solução, os autores desenvolveram um novo esquema de pontuação das fronteiras, visando gerar uma sequência de exploração mais eficiente que a anterior. Além disso, uma abordagem de dois estágios para a geração dessa sequência auxiliou na elaboração e correção da rota, evitando o comportamento de *stop-and-go* do robô. No entanto, a detecção de fronteiras permaneceu inalterada em relação à metodologia FUEL [27].

Em [29], os autores buscaram reduzir a complexidade do mapeamento e otimizar o processo de exploração por meio da criação de um mapa topológico inspirado na estrutura das colmeias de abelhas. Nesse modelo, a expansão do mapa segue regras semelhantes às utilizadas por esses insetos na construção de seus ninhos. A metodologia foi aplicada em ambientes simulados *indoors*, com

o principal objetivo de viabilizar a exploração de áreas afetadas por desastres. Para aumentar a eficiência da exploração, os autores empregaram múltiplos drones operando de forma distribuída, permitindo que a cobertura do ambiente ocorresse de maneira descentralizada e cooperativa. Além da exploração em si, a metodologia também incorporou a detecção de focos de incêndio, utilizando sensores simulados de temperatura. Dessa forma, ao término da missão, o mapa expandido não apenas delimitava as áreas exploradas, mas também indicava a localização dos focos de calor identificados durante a navegação.

O trabalho desenvolvido em [28], obteve bons resultados na exploração utilizando uma abordagem que combina a metodologia clássica de detecção de fronteiras com o potencial das estruturas de mapeamento em OctoMap. Assim como na metodologia FUEL, uma abordagem incremental foi adotada na detecção de fronteiras. Basicamente, as fronteiras analisadas eram provenientes de um sub-mapa formado pelos grids que tiveram seu status modificado na última atualização do mapa. Essa técnica permitiu um aumento da eficiência na detecção de fronteiras. No entanto, para os casos em que a quantidade de fronteiras detectadas aumentou, houve um aumento no custo computacional, o que caracteriza a incerteza quanto ao tempo necessário para detectar as fronteiras. As fronteiras detectadas foram agrupadas em clusters por meio do algoritmo Mean-shift [30]. Para a tomada de decisão, os autores apresentaram uma equação que estima a quantidade de informação a ser obtida caso o robô escolha um determinado cluster. Essa quantidade de informação é determinada pela soma dos grids livres ao redor do cluster, ponderada pela distância entre o cluster e a posição atual do robô. O cluster com a maior pontuação é escolhido como próximo objetivo. A quantidade de grids livres considerados é fixa e também não tem correlação com o tamanho da fronteira, desta maneira, problemas de subestimação de regiões relevantes poderão ocorrer. Por fim, um algoritmo baseado no planejador de caminhos RRT [31] gera um conjunto de pontos que conduzem o robô até a posição objetivo, garantindo a ausência de colisões.

Uma outra abordagem para explorar um ambiente desconhecido é por meio do uso da técnica denominada *Next-Best-View* (NBV), que busca determinar qual será o melhor próximo local para o drone se deslocar, potencialmente resultando

no maior ganho de abertura do mapa. Essa metodologia teve sua origem na geração de perspectivas ótimas para visualização completa de objetos [32], mas sua ideia e metodologia podem ser adaptadas para resolver desafios relacionados à exploração autônoma de drones. Em um estudo conduzido por [33], foi proposta uma abordagem baseada no conceito de NBV, porém sem o uso de fronteiras. Nessa metodologia, uma árvore aleatória (i.e., RRT) é expandida pelos grids livres presentes no ambiente mapeado, e um conjunto de nós é amostrado para criar uma lista de possíveis células candidatas à exploração. A ordenação dessas amostras é determinada por um ganho que é ajustado iterativamente à medida que o robô se desloca em direção à amostra que obteve a maior pontuação. A informação é derivada da quantidade de grids não mapeados que se encontram dentro do campo de visão do sensor quando aquela visão será realizada. Embora essa metodologia tenha demonstrado bons resultados na exploração do mapa, o custo computacional associado à necessidade repetitiva de replanejamento de caminhos é consideravelmente elevado [28].

A metodologia NBV não garante a exploração completa de todos os ambientes e pode resultar na convergência da exploração para mínimos locais [34]. Esse problema não é observado nas estratégias que se baseiam na exploração de fronteiras. Alguns trabalhos recentes optaram por combinar os dois métodos fundamentais, como pode ser visto em [34]. Os autores observaram que as fronteiras são bons indicadores de para onde o robô deve se dirigir para explorar eficientemente o ambiente, ao passo que a abordagem baseada em amostragem (i.e., NBV) elimina a necessidade de pós-processamento de clusterização após a obtenção das fronteiras, reduzindo assim a complexidade computacional da exploração. Adicionalmente, os autores introduziram uma otimização angular que determina a melhor visão em relação aos 360 graus possíveis ao redor do candidato. Esta abordagem foi diretamente comparada ao NBV original proposto por [33], apresentando resultados superiores.

Em [35], os autores introduziram uma metodologia de exploração autônoma para robôs móveis, na qual técnicas anteriormente aplicadas na exploração autônoma de drones foram adaptadas para esse contexto. Essa abordagem versátil transcende as limitações de uma única tecnologia, permitindo que as inovações na exploração

autônoma sejam aplicadas em diferentes cenários. A estratégia de exploração desenvolvida baseia-se em uma topologia hierárquica do mapa baseado em fronteiras, o que possibilita o planejamento de caminhos que direcionam o robô durante a exploração. O algoritmo proposto opera por meio de microexplorações, visando visitar todas as regiões identificadas na primeira iteração do algoritmo. À medida que novas regiões são mapeadas e informações adicionais são extraídas do ambiente, essas áreas são apresentadas ao algoritmo de exploração somente após a conclusão da microexploração. Posteriormente, as novas regiões de exploração são definidas, e o planejador global fornece um novo caminho para a exploração. Durante o processo de exploração, um planejador de caminhos local é empregado para aprimorar o caminho global, aumentando a estabilidade e eficiência da metodologia. Esse refinamento é orientado pelas vistas (i.e., NBV) das regiões a serem exploradas. Para determinar a ordem de visita de cada região planejada, o planejador global utiliza uma metodologia conhecida como *Priority Constrained Asymmetric Traveling Salesman Problem* (PCATSP). A característica de microexplorações mantém a dimensionalidade do problema baixa, viabilizando a solução do algoritmo PCATSP dentro de um tempo razoável. O trabalho foi comparado às metodologias FAEP e FUEL, e atingiu a exploração volumétrica máxima do ambiente em um menor tempo.

Em [36], foi proposta uma nova abordagem para a detecção de fronteiras, utilizando um método amostral baseado no algoritmo RRT. A exploração emprega uma heurística inspirada nos movimentos de ondas, onde a detecção de fronteiras ocorre pela análise da presença de um nó amostrado em uma região que pode ser caracterizada como fronteira. Essa estratégia, aplicada em ambientes 2D, visa localizar rapidamente as fronteiras no ambiente, contornando assim o desafio da busca completa no mapa. Os autores adotam a metodologia de clusterização Mean-shift, similar àquela observada em [28] para agrupar as fronteiras, enquanto a tomada de decisão é conduzida por meio da solução do problema FSOTSP (*Fixed Start Open Travelling Salesman Problem*) usando o algoritmo genético [37]. Além do custo total associado à visita de todas as fronteiras, a solução do FSOTSP também leva em consideração a quantidade de informação que pode ser obtida nos grids ao redor da fronteira, bem como a distância euclidiana. No entanto,

as desvantagens dessa metodologia incluem a ausência de garantia na obtenção de todas as fronteiras e o tempo excessivo necessário para explorar a árvore em cenários mais amplos.

A metodologia de exploração apresentada em [38] focou na construção otimizada dos mapas a serem explorados. Um dos desafios comuns na exploração é garantir a abertura completa do mapa, pois pequenas áreas no ambiente podem permanecer fechadas devido ao seu baixo peso em relação à quantidade de informação explorável em outras regiões do mapa, resultando no isolamento dessas pequenas fronteiras. Os autores propuseram uma estratégia controlada, na qual as melhores fronteiras selecionadas durante a exploração são armazenadas para serem consideradas em futuras tomadas de decisão. Eles alcançaram uma taxa de volume explorado superiores a outras metodologias de exploração autônoma. No entanto, para alcançar esses resultados, o tempo de exploração e a distância percorrida foram superiores em comparação com outras abordagens de exploração autônoma. O processo de detecção de fronteiras adotou a metodologia proposta por [28], reconhecida por sua eficiência computacional. Contudo, essa abordagem apresenta uma limitação inerente: a impossibilidade de prever a quantidade de fronteiras que emergirão durante a missão exploratória. Essa indeterminação quantitativa implica em uma complexidade temporal variável, podendo resultar em uma sobrecarga computacional e também uma incerteza operacional.

Diante da revisão bibliográfica apresentada nesta seção, é possível observar a importância da proposição de novas metodologias de exploração autônoma. Um problema comum observado na literatura é a complexidade computacional na busca por fronteiras nos mapas. Dessa forma, esta pesquisa busca por alternativas em relação à metodologia seminal [21] que tornará a missão autônoma plausível a partir de um tempo factível dada as restrições energéticas que os robôs possuem. Portanto, o trabalho de pesquisa proposto nessa tese objetiva o desenvolvimento de uma metodologia para a extração de fronteiras que não terá dependência direta da quantidade de fronteiras utilizando *deep learning*. O resultado desta inteligência também será incorporado à tomada de decisão para que o robô seja guiado para as melhores fronteiras elegíveis de serem selecionadas.

## 2.1 METODOLOGIA PROPOSTA $\times$ VSLAM

É fundamental destacar nesta revisão bibliográfica as diferenças metodológicas entre as técnicas convencionais de *Visual Simultaneous Localization and Mapping* (VSLAM) e a abordagem proposta neste trabalho evidenciada em 1.1 para que seja possível entender como esta importante e consolidada técnica pode ser agregada ao que será proposto por esta tese. Como demonstrado por [39], é possível estimar a posição cartesiana 3D utilizando exclusivamente uma câmera monocular, onde o sistema emprega *Filtros de Kalman Estendidos* (EKF, do inglês *Extended Kalman Filter*) para correlacionar observações visuais sequenciais com a estimativa de pose e construção do mapa. Contudo, as técnicas clássicas de VSLAM concentram-se primordialmente na dualidade localização-mapeamento, sem incorporar mecanismos intrínsecos de tomada de decisão ou interpretação semântica do ambiente. Em particular, a metodologia original não contempla a identificação de fronteiras ou regiões navegáveis, limitando-se à reconstrução geométrica passiva do espaço.

Os primeiros sistemas de VSLAM representaram um marco significativo na robótica e visão computacional, demonstrando a viabilidade do mapeamento simultâneo e localização utilizando apenas uma câmera monocular. Contudo, essas implementações pioneiras apresentaram dois desafios fundamentais: (i) a acumulação progressiva de erros de localização (*drift*), que comprometia a precisão geométrica do mapa reconstruído, e (ii) a inerente ambiguidade escalar, decorrente da impossibilidade de determinar métricas absolutas a partir de fluxo óptico monocular. Essas limitações tornavam-se particularmente críticas em trajetórias extensas ou quando a magnitude do deslocamento da câmera era incerta. Em resposta a esses desafios, surgiram abordagens inovadoras que preservaram a premissa da câmera monocular como sensor principal, porém incorporando técnicas avançadas de otimização e restrições geométricas para mitigar esses efeitos.

Os métodos proeminentes de SLAM baseados em *features*, em particular a família ORB-SLAM (*Oriented FAST and Rotated BRIEF SLAM*) [40] - incluindo suas versões ORB-SLAM2 [41] e ORB-SLAM3 [42] -, fundamentam-se na detecção eficiente de pontos-chave invariantes a rotação e escala em frames consecutivos. Esses descritores ORB combinam o detector *Oriented FAST* com o descritor

*Rotated BRIEF*, garantindo extração e correspondência rápidas mesmo em sistemas computacionalmente limitados.

Enquanto a versão original [40] utilizava câmeras monoculares, o ORB-SLAM2 expandiu significativamente suas capacidades ao adicionar suporte a câmeras estéreo e RGB-D, resolvendo assim o problema da ambiguidade escalar inerente a sistemas monoculares. O ORB-SLAM3, por sua vez, deu um passo adiante ao integrar dados inerciais de IMUs (*Inertial Measurement Units*), mitigando drifts durante movimentos bruscos e permitindo inicialização imediata do mapa. Esses avanços não apenas ampliaram a aplicabilidade a cenários complexos, como ambientes dinâmicos ou com condições luminosas variáveis, mas também elevaram significativamente a precisão da localização.

Em contraste, as técnicas propostas neste trabalho estendem as capacidades tradicionais do VSLAM, integrando mapeamento autônomo com tomada de decisão em tempo real. Enquanto sistemas convencionais de VSLAM limitam-se à reconstrução geométrica e estimativa de pose, nossa metodologia avança esse paradigma ao incorporar dois componentes fundamentais.

Primeiramente, introduzimos um módulo de detecção de fronteiras navegáveis, capaz de discriminar regiões acessíveis de obstáculos intransponíveis com base na análise geométrica e semântica do ambiente. Essa capacidade é essencial para aplicações em ambientes não estruturados, onde a delimitação de espaços seguros não pode ser pré-definida.

Em segundo lugar, propomos uma estrutura hierárquica de decisão que interpreta dinamicamente o mapa gerado pelo VSLAM, transformando dados brutos em comandos de navegação. Esse sistema não apenas reconstrói o ambiente, mas também planeja trajetórias otimizadas considerando restrições físicas e objetivos operacionais.

Essa abordagem é particularmente relevante para cenários que demandam autonomia completa, como inspeções de plantas industriais ou missões de busca em ambientes catastróficos. Diferentemente das soluções passivas de VSLAM, nosso *framework* fecha o ciclo percepção, decisão e ação, permitindo que o robô não apenas compreenda o ambiente, mas também interaja com ele de forma inteligente e proposital. Sendo assim, é fácil notar que a metodologia proposta por esta tese

se estende como uma ferramenta às aplicações que utilizam VSLAM.

### 3 FUNDAMENTOS TEÓRICOS

Esta seção visa introduzir os conceitos teóricos fundamentais de todos os algoritmos e metodologias que servirão como base para o trabalho proposto nesta tese. A interação entre esses algoritmos será detalhada na Seção 4, cujo objetivo principal é apresentar a missão de exploração autônoma proposta.

#### 3.1 METODOLOGIA PARA DETECÇÃO DE OBJETOS - YOLO

##### 3.1.1 Introdução à YOLO

Esta seção tem como objetivo apresentar a técnica que será utilizada para analisar imagens geradas por uma nuvem de pontos que gera o mapa do ambiente, visando encontrar e localizar todas as fronteiras presentes no ambiente durante a exploração. Dessa forma, essa busca se configura como um problema de detecção, uma vez que, além de encontrar (i.e, detectar) o objeto (i.e., fronteira), é necessário identificar a região exata onde a fronteira está localizada na imagem. Essa localização proporcionará ao robô as informações necessárias para que as missões de exploração do ambiente sejam executadas.

Esse tipo de problema acrescenta complexidade à implementação, pois as metodologias comuns de classificação, que empregam redes neurais convolucionais (do inglês, *Convolutional Neural Networks* - CNNs) para detectar a presença do objeto na imagem, não são suficientes para superar esse desafio, devido à quantidade variável de objetos que podem estar presentes em uma imagem. Um aspecto crucial nas tarefas de detecção de objetos é a sua generalização, já que as características que os definem podem ser extremamente diversas, como cores, tamanhos e formatos, sendo esses fatores determinantes para alcançar uma detecção precisa de objetos.

Para compreender a motivação por trás da existência da rede neural *You Only Look Once* (YOLO), é essencial apresentar as metodologias de detecção de objetos anteriores a 2015, ano de criação da YOLO. Isso proporcionará uma visão mais completa do contexto e da evolução das técnicas utilizadas na detecção de objetos, fundamentando a escolha da YOLO como uma abordagem inovadora e mais eficiente.

Em 2010, [43] apresentou um trabalho de grande relevância nas tarefas de classificação e detecção de objetos. A metodologia, chamada de *Deformable Parts Models* (DPM), utiliza uma abordagem baseada em janelas deslizantes (*sliding window*) sobre imagens. Essa janela é deslocada ao longo da imagem em espaçamentos uniformes. O DPM segmenta os modelos a serem detectados em partes e, por meio de técnicas que envolvem uma variante de *Support Vector Machine* (SVM), realiza a predição da presença ou ausência do objeto na imagem.

A avaliação é determinada pela pontuação atribuída à combinação entre o modelo raiz (objeto inteiro) e o modelo em partes com uma resolução maior que o primeiro. Esses modelos são caracterizados por filtros do tipo HOG (*Histograms of Oriented Gradient* [44]). A aplicação do aspecto deformável nas partes proporciona uma capacidade de generalização no reconhecimento dos objetos. Entretanto, o custo computacional da metodologia, devido à quantidade de classificadores SVM em cada parte segmentada, à aplicação do filtro de HOG e à necessidade de um banco de dados extenso e detalhado para treinamento, são fatores que levaram os pesquisadores a continuarem explorando novas aplicações na detecção de objetos. Esta ineficiência inerente ao método impacta diretamente na aplicação desta metodologia em tempo real, desta forma, melhorias no DPM puderam aplicar essa metodologia em taxas de 30 *Hz* e 100 *Hz* estes resultados foram publicados no trabalho apresentado em [45].

Devido ao enorme sucesso das redes neurais convolucionais [46] na extração de características de imagens, diversos trabalhos em detecção de objetos têm surgido com base nesse tipo de arquitetura. A principal vantagem de utilizar a CNN é a dispensa de atividades de pré-processamento das imagens, como a implementação de filtros, uma vez que durante a fase de treinamento os filtros convolucionais são capazes de desempenhar este papel.

No entanto, empregar uma rede CNN para a tarefa de detecção de objetos é desafiador, pois a quantidade de objetos a ser analisado varia consideravelmente em relação à saída de um modelo de CNN treinado. Além disso, substituir os classificadores SVM da metodologia DPM também não é vantajoso, já que a quantidade de janelas a serem classificadas é grande, resultando em um elevado custo computacional para realizar tal tarefa.

Em 2013, o trabalho pioneiro de Girshick et al., denominado Region-based CNN (R-CNN) [47], introduziu o uso eficiente de redes neurais convolucionais (CNNs) para detecção de objetos. A abordagem baseava-se na análise de um número finito de regiões de interesse (cerca de 2 mil por imagem), extraídas pelo algoritmo de busca seletiva proposto em [48]. Cada região era processada por uma CNN para extração de características, seguidas por classificação via SVM. O SVM não apenas determinava a presença do objeto na caixa delimitadora, mas também ajustava suas coordenadas por meio de quatro parâmetros de regressão. No entanto, apesar da inovação, o método apresentava limitações críticas, como alto custo computacional e tempo excessivo de treinamento, devido à necessidade de processar cada região individualmente.

Dada a boa qualidade na acurácia da metodologia R-CNN, foram propostos algoritmos com o objetivo de reduzir o tempo necessário para classificação e treinamento, tais como o *Fast R-CNN* [49] e o *Faster R-CNN* [50]. O primeiro método eliminou a necessidade de usar uma CNN independente para cada uma das regiões de interesse. Agora, a imagem inteira é passada para a CNN, e a partir do mapa de características obtido, é aplicada uma camada de *pooling* nas regiões de interesse. A saída dessa camada é transformada em um vetor, onde é possível realizar a predição da classe e os parâmetros da caixa delimitadora a serem atualizados. A grande vantagem dessa metodologia é que não é necessário realizar a classificação em 2000 regiões de interesse, sendo feita apenas uma vez. Conforme demonstrado pelos autores, a metodologia apresentada ainda possui ineficiência computacional devido à consideração de regiões de interesse ao longo do processo.

Como melhoria, a metodologia *Faster R-CNN* propôs a substituição do algoritmo de busca seletiva das regiões de interesse por uma camada denominada *region proposal network* (RPN) para prever essas regiões. Os resultados alcançados no trabalho foram satisfatórios, uma vez que o tempo para a detecção de um objeto na imagem diminuiu drasticamente em comparação com o *Fast R-CNN* e o R-CNN. No entanto, essa técnica não se saiu bem ao executar tarefas em tempo real [51].

Diante das dificuldades enfrentadas na execução de tarefas de detecção de objetos em tempo real, pesquisas subsequentes observaram que a abordagem com regiões de interesse aumentava a complexidade computacional. Assim, em 2015,

foi apresentada uma metodologia denominada YOLO (*You Only Look Once*), cuja tradução para o português é "você olha somente uma vez". Esse nome revela como a análise da imagem se desencadeará, uma vez que a detecção de objetos será tratada como um problema de regressão, ou seja, uma passagem pela arquitetura retornará a probabilidade do objeto encontrado e a caixa delimitadora ao redor do mesmo. Essa característica proporcionou um considerável avanço na eficiência do tempo para identificação e localização de objetos. Até o momento da publicação do YOLO, todos os métodos envolviam a inferência em dois estágios: a definição da caixa delimitadora e a classificação do objeto. Agora, com o advento dessa metodologia YOLO, ambos os estágios são realizados simultaneamente.

De acordo com os autores, essa nova abordagem de analisar imagens resultou em três grandes benefícios diretos:

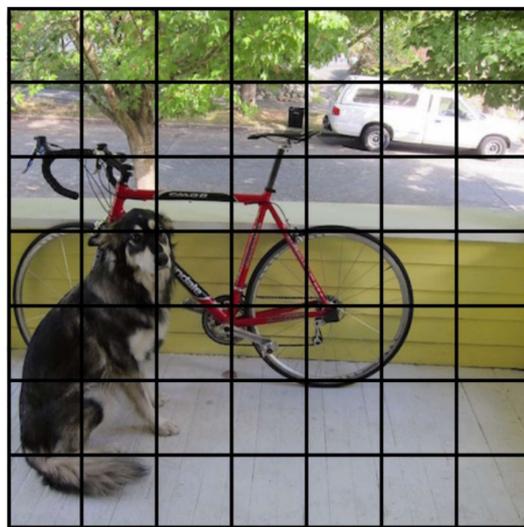
- Detecção extremamente rápida: A YOLO se destaca pela sua velocidade, comparada ao estado da arte da época, devido ao seu enfoque como um problema de regressão;
- Análise da imagem inteira durante as predições: Diferentemente das metodologias anteriormente mencionadas, a YOLO treina a imagem como um todo, não apenas as regiões de interesse. Essa abordagem possibilita obter resultados mais robustos a partir de um banco de dados mais genérico;
- Generalização na aprendizagem de objetos: Graças à sua arquitetura, a YOLO consegue aprender de maneira generalizada o que constitui um objeto, tornando-a menos suscetível a variações nas características do objeto.

Os métodos de detecção de objetos em dois estágios, como o R-CNN, Fast R-CNN e Faster R-CNN, são amplamente reconhecidos por sua alta acurácia. Contudo, a complexidade inerente à necessidade de processar múltiplas regiões de interesse resulta em uma demanda computacional elevada, tornando esses métodos menos viáveis para aplicações em tempo real. Em contraste, a metodologia YOLO resulta em uma detecção extremamente rápida e eficiente, ao custo de uma leve perda de acurácia comparada aos métodos de dois estágios.

### 3.1.2 Modelagem YOLO

O processo de detecção de objetos no *YOLO* inicia-se com a divisão da imagem em uma grade (*grid*) regular de dimensões  $S \times S$ . Na implementação seminal [51], os autores propuseram  $S = 7$ , conforme ilustrado na Figura 3. Cada célula da grade é responsável pela predição de  $B$  conjuntos de parâmetros. Na primeira versão do algoritmo, utilizou-se  $B = 2$  conjuntos de parâmetros por célula, permitindo a detecção de múltiplos objetos centralizados na mesma região.

Figura 3 – Divisão em *Grid* de Uma Imagem.



Fonte: Retirado de [51].

O conjunto de parâmetros pode ser representado por um vetor  $\mathbf{p} = (x, y, w, h, c) \in \mathbb{R}^5$ , onde:

- $(x, y)$  denotam as coordenadas normalizadas do centro da caixa delimitadora em relação à célula da grade;
- $(w, h)$  representam a largura e altura da caixa, normalizadas pela dimensão da imagem;
- $c$  corresponde à confiança da detecção, definida como:

$$c = Pr(\text{Objeto}) \times \text{IoU}_{\text{pred}}^{\text{truth}} \quad (3.1)$$

sendo que  $Pr(\text{Objeto}) \in \{0, 1\}$  indica a presença (1) ou ausência (0) do objeto, e  $\text{IoU}_{\text{pred}}^{\text{truth}}$  representa a interseção sobre união (*Intersection over Union*) entre a predição e a região verdadeira (*ground truth*). Durante o treinamento, os pesos do modelo são ajustados para maximizar essa medida de confiança, melhorando progressivamente a similaridade entre as predições e os objetos reais. Embora cada célula produza  $B = 2$  predições, apenas aquela com maior valor de confiança é considerada na saída final do modelo.

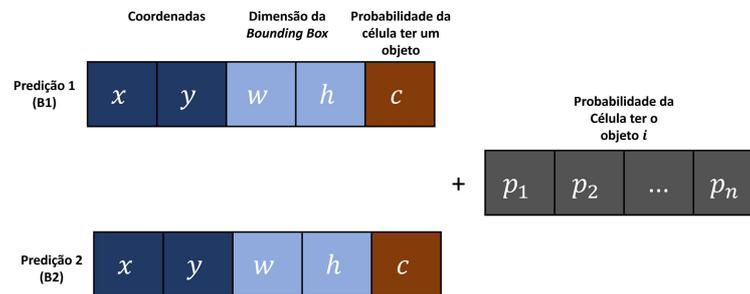
Além dos cinco parâmetros fundamentais  $(x, y, w, h, c)$ , a YOLO prevê probabilidades condicionais  $P(\text{Class}_i | \text{Object})$  para cada uma das  $N = 20$  classes do PASCAL VOC 2012 [52]. Estas probabilidades formam um vetor  $\mathbf{P} \in \mathbb{R}^N$  normalizado por softmax, onde cada componente  $P(\text{Class}_i | \text{Object})$  representa a probabilidade da caixa conter um objeto da  $i$ -ésima classe, dado que um objeto está presente.

As predições para detecção de objetos são organizadas em um tensor tridimensional para cada divisão da imagem. No caso específico da arquitetura YOLO original, o tensor de saída possui dimensões  $7 \times 7 \times 30$ , onde:

- As duas primeiras dimensões ( $7 \times 7$ ) correspondem às células da grade espacial;
- A terceira dimensão (30) resulta da concatenação de:
  - $B = 2$  conjuntos de parâmetros de *bounding boxes* (5 parâmetros cada:  $x, y, w, h, c$ ), totalizando 10 valores;
  - 20 probabilidades condicionais  $P(\text{Class}_i | \text{Object})$  para as classes do PASCAL VOC 2012.

A estrutura completa deste tensor de saída é ilustrada na Figura 4, que detalha a organização espacial e a composição das camadas de características.

Figura 4 – Estrutura de Saída com  $B = 2$  e  $N$  Classes.



Elaborado pelo Autor.

### 3.1.3 Arquitetura de Rede YOLO

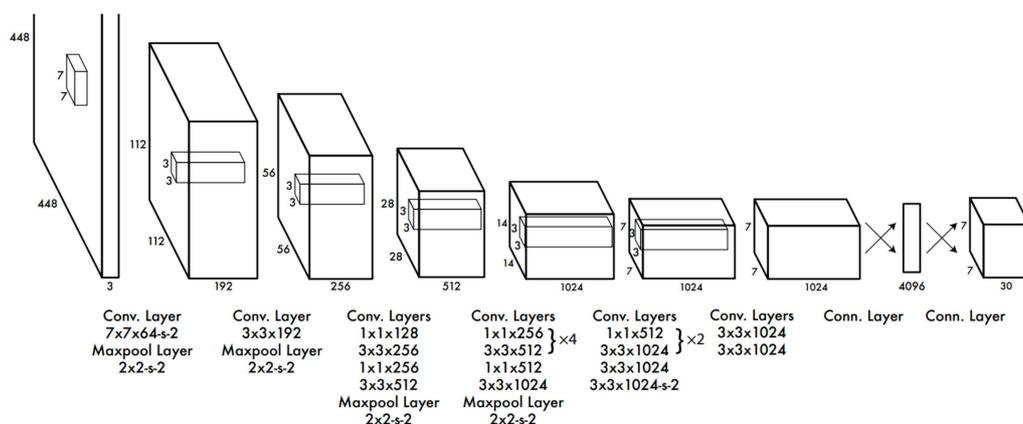
A arquitetura da rede neural empregada no YOLO é baseada no modelo GoogLeNet [53], originalmente desenvolvido para o desafio *ImageNet Large-Scale Visual Recognition Challenge* (ILSVRC) de 2014. A estrutura da rede compreende três componentes principais:

- **Camadas convolucionais:** Extraem e compõem mapas de características hierárquicos a partir da imagem de entrada, capturando padrões espaciais em diferentes escalas;
- **Camadas de *max-pooling*:** Reduzem dimensionalidade espacial, preservando as características mais ativas em cada região ( $\max(x_{i,j})$ ), conferindo invariância a pequenas translações e variações na entrada;
- **Camadas *fully connected*:** Combinam linearmente todas as características aprendidas (via produto matricial  $\mathbf{W}\mathbf{x} + \mathbf{b}$ ) utilizando os pesos  $\mathbf{W}$  otimizados durante o treinamento, sintetizando a informação para a tarefa de detecção.

Esta arquitetura híbrida permite que as operações iniciais (convoluções e *poolings*) extraiam características visualmente significativas, enquanto as camadas densas (*fully connected*) aprendem combinações não-lineares desses padrões para a classificação e localização precisa de objetos.

A arquitetura da YOLO versão 1, inspirada no GoogLeNet [53], é composta por uma sequência de 24 camadas convolucionais intercaladas por 3 operações de *max-pooling*, seguidas por uma camada *fully connected* final. Esta configuração foi cuidadosamente projetada para produzir o tensor de saída  $7 \times 7 \times 30$ . Além da arquitetura completa, a Figura 5 apresenta a dimensão dos filtros e a progressão dimensional da rede implementada.

Figura 5 – Estrutura da Rede Neural da YOLO.



Fonte: Retirado de [54].

Além dos componentes já discutidos, observa-se que determinadas camadas convolucionais empregam *strides* superiores a 1. Formalmente, o *stride*  $s \in \mathbb{Z}^+$  define o deslocamento espacial do kernel durante a operação de convolução, conforme a expressão:

$$(f * g)[i, j] = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f[m, n] \cdot g[i - sm, j - sn] \quad (3.2)$$

onde  $f$  representa o mapa de características de entrada e  $g$  o kernel convolucional. Esta configuração possui implicações fundamentais:

- **Eficiência computacional:** Valores maiores de  $s$  reduzem quadraticamente as operações necessárias, diminuindo o custo computacional. Para um *stride*  $s$ , a redução é proporcional a  $s^2$  na dimensão espacial;

- **Redução dimensional:** Para uma entrada de dimensões  $H_{in} \times W_{in}$ , o *stride*  $s > 1$  reduz as dimensões de saída conforme:

$$\begin{cases} H_{out} = \left\lfloor \frac{H_{in} + 2p_h - k_h}{s} \right\rfloor + 1 \\ W_{out} = \left\lfloor \frac{W_{in} + 2p_w - k_w}{s} \right\rfloor + 1 \end{cases} \quad (3.3)$$

onde:

- $k_h \times k_w$ : dimensões do *kernel* (vertical  $\times$  horizontal);
  - $p_h \times p_w$ : dimensões do *padding* (vertical  $\times$  horizontal) que adiciona zeros nas bordas da entrada para controlar o tamanho da saída. A informação do *padding* é suprimida na Figura 5, porém, estes valores existem para que seja mantida a proporção espacial entre as camadas.
- **Compromisso representacional:** Embora *strides* maiores acelerem o processamento, eles induzem perda de informação espacial fina, limitando a capacidade de detectar características locais de dimensões pequenas;

Este *trade-off* entre eficiência e sensibilidade a detalhes é particularmente relevante na detecção de objetos pequenos, onde a preservação de informações espaciais é crucial.

A arquitetura YOLO opera com imagens de entrada na resolução fixa de  $448 \times 448$  pixels, que de acordo com autores, é uma escolha que equilibra a captura de características espaciais detalhadas com a viabilidade computacional. Essa resolução superior permite a detecção de objetos pequenos e preserva informações de alta frequência, sendo particularmente adequada para a estrutura de grade  $7 \times 7$  da camada final. Durante o pré-treinamento no conjunto ImageNet [55] com aproximadamente mil classes, utilizou-se a resolução reduzida de  $224 \times 224$  pixels para acelerar a convergência inicial e manter compatibilidade com arquiteturas de classificação existentes, seguida por um ajuste fino na resolução completa.

Como evidenciado na Figura 5, a YOLO inovou ao formular a detecção como um problema de regressão unificada  $f : \mathbb{R}^{448 \times 448 \times 3} \rightarrow \mathbb{R}^{7 \times 7 \times 30}$ , transformando radicalmente o paradigma de detecção de objetos. Essa abordagem *end-to-end*

elimina a necessidade de arquiteturas complexas com estágios independentes, permitindo otimização conjunta de todos os parâmetros e aproveitamento superior do contexto global da imagem, enquanto mantém eficiência computacional para aplicações em tempo real [51].

### 3.1.4 Treinamento da Rede YOLO

A arquitetura YOLO emprega pesos pré-treinados obtidos durante a fase inicial de treinamento desenvolvida pelos autores, utilizando o banco de dados ImageNet [55]. Este conjunto de dados compreende 1.000 classes distintas de objetos, com 1.2 milhão de imagens para treinamento e 50.000 para validação, originalmente preparado para o desafio ILSVRC. Nesta etapa preliminar, o objetivo principal é desenvolver uma estrutura com capacidade discriminativa robusta para identificação de objetos, sem ainda considerar sua localização espacial nas imagens. Esse processo de pré-treinamento em classificação permite que a rede aprenda características visuais hierárquicas que serão posteriormente adaptadas para a tarefa específica de detecção de objetos durante a fase de ajuste fino (*fine-tuning*), garantindo tanto a generalização quanto a especialização do modelo.

Para esta tarefa de pré-treinamento, adotou-se uma arquitetura derivada daquela apresentada na Figura 5, porém com duas modificações fundamentais: remoção das últimas 4 camadas convolucionais (reduzindo de 24 para 20 camadas) e eliminação das camadas *fully connected*, mantendo apenas as operações convolucionais e de *pooling*. Adicionalmente, a resolução de entrada foi reduzida para  $224 \times 224$  pixels durante esta fase, visando reduzir o custo computacional, manter compatibilidade com as dimensões padrão do ImageNet e permitir uma inicialização mais eficiente dos pesos. Esta configuração simplificada é otimizada para a tarefa de classificação de imagens, enquanto a arquitetura completa com resolução de  $448 \times 448$  pixels será utilizada posteriormente para a detecção de objetos.

Na modificação arquitetural proposta, a remoção das camadas finais da rede original é acompanhada pela inserção de uma camada de *average-pooling* global seguida de uma camada *fully connected* com dimensionalidade  $N_c$ , onde  $N_c$  corresponde ao número de classes no conjunto de dados. Esta abordagem difere fundamentalmente da operação de *max-pooling* tradicionalmente empregada, pois

enquanto o *max-pooling* ( $\mathcal{M}$ ) preserva apenas os ativadores mais proeminentes através da operação  $\mathcal{M}(X) = \max_{i,j}(X_{i,j})$  para cada região  $X$  da *feature map*, o *average-pooling* ( $\mathcal{A}$ ) realiza uma transformação suavizante calculando  $\mathcal{A}(X) = \frac{1}{|X|} \sum_{i,j} X_{i,j}$ , o que proporciona:

- Maior robustez a variações espaciais nos padrões de ativação;
- Incorporação de informações de toda a região receptiva;
- Redução de *overfitting* pela integração de informações distribuídas.

Esta configuração é particularmente adequada para tarefas de classificação, onde a preservação de informações contextuais globais é prioritária sobre a localização espacial precisa de características.

A arquitetura emprega funções de ativação específicas para cada estágio: nas camadas intermediárias, utiliza-se a função *LeakyReLU* com coeficiente  $\alpha = 0.1$ , definida por  $\text{LeakyReLU}(x) = \max(x, \alpha x)$ , que preserva um fluxo de gradiente suave ( $\alpha = 0.1$ ) para entradas negativas, evitando o problema de neurônios inativos típico da ReLU convencional. Na camada final, adota-se uma função linear  $\phi(x) = x$ , cuja saída não limitada é essencial para a tarefa de regressão, permitindo que a rede gere coordenadas arbitrárias para as *bounding boxes* e valores de probabilidade não restritos. Esta combinação estratégica de funções de ativação equilibra a aprendizagem de características nas camadas ocultas com a flexibilidade necessária na saída da rede.

Essa estratégia de pré-treinamento, originalmente proposta em [56], emprega uma abordagem de aprendizado por transferência em duas fases distintas. As camadas convolucionais inferiores são inicializadas com pesos pré-treinados em um conjunto de dados genérico (como o ImageNet), capturando características visuais básicas. Para as quatro camadas convolucionais superiores adicionadas - que devem aprender características específicas das novas classes - os pesos são inicializados de maneira aleatória.

A segunda fase consiste no treinamento da rede neural com o banco de dados específico contendo os objetos a serem identificados. Redmon, autor da

YOLO, optaram por utilizar a função de soma do erro quadrático (SSE, do inglês *sum of squared errors*), conforme a Equação 3.4, como métrica para quantificar a discrepância entre as predições da rede e os valores esperados.

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.4)$$

Na formulação apresentada,  $n$  representa a quantidade total de predições realizadas durante o treinamento para cada amostra  $i$ , sendo  $y_i$  o valor predito pela rede e  $\hat{y}_i$  o valor esperado correspondente. A escolha desta função de perda se justifica por sua propriedade de diferenciabilidade em todo o domínio, o que permite a aplicação eficaz de métodos de otimização baseados em gradiente descendente para ajuste dos pesos da rede. Além disso, a natureza quadrática da SSE confere maior sensibilidade a erros de predição mais significativos, direcionando o processo de aprendizagem a priorizar a correção dessas discrepâncias.

Como estabelecido anteriormente, a arquitetura YOLO produz como saída dois conjuntos de predições (totalizando 10 parâmetros:  $x, y, w, h, c$  para cada *bounding box*) além das probabilidades de classe. Entretanto, durante o treinamento, os autores identificaram uma instabilidade decorrente do desbalanceamento natural entre células contendo objetos e células vazias (*background*). Essa disparidade, onde a razão entre células sem objetos e com objetos é significativamente maior que 1, faz com que as predições de confiança zero dominem o cálculo do gradiente, distorcendo as atualizações dos parâmetros e prejudicando a convergência do modelo.

Para resolver este problema, os autores implementaram uma ponderação diferenciada na função de perda, introduzindo constantes específicas:  $\lambda_{\text{coord}}$  para os erros de localização e  $\lambda_{\text{noobj}}$  para células sem objetos. Essa abordagem reduz a influência desproporcional das predições negativas enquanto mantém o foco na precisão das coordenadas dos objetos, estabilizando significativamente o processo de treinamento. A função de perda resultante ( $SSE_{\text{YOLO}}$ ) incorpora esses fatores de ponderação de maneira sistemática, como será detalhado na Equação 3.5.

$$\begin{aligned}
SSE_{YOLO} = & \lambda_{coord} \sum_{i=0}^{S \times S} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
& + \lambda_{coord} \sum_{i=0}^{S \times S} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\
& + \sum_{i=0}^{S \times S} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 \\
& + \lambda_{noobj} \sum_{i=0}^{S \times S} \sum_{j=0}^B \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\
& + \sum_{i=0}^{S \times S} \mathbb{I}_{ij}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
\end{aligned} \tag{3.5}$$

Neste contexto,  $S \times S$  representa a quantidade total de *grids* em que a imagem é dividida, enquanto  $B$  indica o número de caixas delimitadoras (regiões de interesse) previstas por *grid*. As variáveis  $(x, y)$  correspondem às coordenadas do centro da caixa delimitadora, e  $w$  e  $h$  representam, respectivamente, sua largura e altura. A variável  $C$  refere-se à confiança de que uma determinada célula do *grid* contém um objeto — definida, neste caso, como a probabilidade de presença do objeto multiplicada pela acurácia da caixa delimitadora com relação à região de interseção. Por fim,  $p(c)$  indica a probabilidade de que a caixa contenha um objeto da classe  $c$ . As variáveis preditas são denotadas pelo símbolo ‘^’ (acento circunflexo) sobre a variável correspondente.

É importante destacar que os valores correspondentes à largura ( $w$ ) e à altura ( $h$ ) das caixas delimitadoras são submetidos à operação de raiz quadrada antes do cálculo do erro quadrático. Essa transformação tem como objetivo atenuar o impacto de erros absolutos em regiões de interesse maiores, evitando que pequenos desvios em caixas grandes sejam penalizados com a mesma severidade que erros de igual magnitude em caixas menores. Dessa forma, busca-se uma contribuição mais equilibrada na função de perda, proporcional à escala da região de interesse.

Para capacitar a rede a reconhecer corretamente as regiões de interesse, estabelece-se que apenas uma caixa delimitadora será responsável por representar a predição em cada célula do *grid*. A escolha da caixa é baseada na que apresenta o maior valor de *Intersection over Union* entre as  $B$  caixas preditas naquela célula.

A região  $j$  da célula  $i$  selecionada como responsável pela predição é indicada pela função indicadora  $\mathbb{I}_{ij}^{obj}$ , cujo comportamento está descrito na Equação 3.6. As demais caixas delimitadoras, que não são responsáveis por detectar objetos — seja por ausência de objetos ou por apresentarem menor IoU — são tratadas pela função indicadora  $\mathbb{I}_{ij}^{noobj}$ . Essas contribuições aparecem na quarta linha da Equação 3.5, sendo ponderadas pela constante de penalização  $\lambda_{noobj}$ , definida pelos autores da YOLO como 0.5. Já a constante  $\lambda_{coord}$ , responsável por ponderar os termos relacionados à localização das caixas, assume o valor de 5.

$$\mathbb{I}_{ij}^{obj} = \begin{cases} 1, \text{ Objeto presente no } grid \\ 0, \text{ Objeto ausente no } grid \end{cases} \quad (3.6)$$

$$\mathbb{I}_{ij}^{noobj} = \begin{cases} 1, \text{ Objeto ausente no } grid \\ 0, \text{ Objeto presente no } grid \end{cases} \quad (3.7)$$

Com o objetivo de aprimorar os resultados de predição, além da arquitetura previamente descrita, foram incorporadas diversas técnicas complementares. Entre elas, destaca-se a divisão dos dados em *batches* durante o treinamento, o ajuste dinâmico da taxa de aprendizado ao longo das épocas, o uso de *dropout* — técnica que consiste na inibição aleatória de neurônios durante o treinamento, a fim de mitigar o *overfitting* — e o aumento artificial do conjunto de dados (*data augmentation*), visando ampliar a variabilidade e a robustez do modelo frente a diferentes situações.

Com o objetivo de reduzir o tempo de resposta na detecção de objetos em imagens e viabilizar predições em tempo real de forma eficiente, no mesmo trabalho que foi apresentado a YOLO, os autores, propuseram uma abordagem menor para a arquitetura da rede. Em vez das 24 camadas convolucionais originalmente empregadas para a extração de características, foram utilizadas apenas 9 camadas nessa etapa. Essa modificação resultou em um aumento expressivo na taxa de detecção, que passou de 45 Hz para 155 Hz. No entanto, esse ganho de desempenho veio acompanhado de uma queda nos valores de *Mean Average Precision* (mAP), com uma redução de aproximadamente 17%. Essa versão otimizada da arquitetura foi denominada *Fast YOLO* [51].

Todo o processo de treinamento e teste foi conduzido por Redmon utilizando imagens do conjunto de dados PASCAL VOC 2012 [52] na apresentação da YOLO. As comparações com outras metodologias, conforme apresentado em [51], estão disponíveis na Tabela 1.

Tabela 1 – Metodologias de Detecção de Objetos [51].

Metodologia	Dois Estágios	mAP	Frequência (Hz)
<b>Fast R-CNN</b> [49]	Sim	70.0	0.5
<b>Faster R-CNN</b> [50]	Sim	73.2	7
<b>DPM 30 Hz</b> [45]	Sim	26.1	30
<b>YOLO</b> [51]	Não	63.4	45
<b>DPM 100 Hz</b> [45]	Sim	16.0	100
<b>Fast YOLO</b> [51]	Não	52.7	155

Observa-se que as metodologias baseadas em abordagens de dois estágios apresentaram valores de mAP superiores aos alcançados pelo YOLO. No caso mais expressivo, a arquitetura Faster R-CNN obteve um desempenho aproximadamente 15% superior ao do YOLO. Tal resultado pode ser atribuído, em parte, ao elevado número de regiões de interesse geradas pela Faster R-CNN — cerca de 2000 por imagem — em contraste com as 98 produzidas pelo YOLO, o que aumenta a probabilidade de gerar regiões com alta sobreposição (IoU) em relação às caixas reais (*ground truth*). No entanto, conforme relatado pelos autores, com uma taxa de apenas 7 Hz, esse algoritmo não se mostra adequado para aplicações de detecção em tempo real.

Os resultados obtidos pela YOLO foram notavelmente superiores àqueles reportados por trabalhos anteriores na literatura. No entanto, a metodologia ainda apresentava limitações significativas. A primeira delas refere-se à atribuição de pesos iguais para erros de localização, independentemente do tamanho da região de interesse. Essa abordagem pode gerar imprecisões na detecção, especialmente quando comparando objetos grandes e pequenos. Além disso, a YOLO demonstrou dificuldades em identificar objetos pequenos quando estes aparecem agrupados em uma mesma região. Por fim, a limitação no número de regiões de interesse propostas pelo modelo impõe restrições à sua capacidade de localizar objetos com precisão, comprometendo o desempenho em cenários mais complexos.

### 3.1.5 Versões Baseadas na YOLO

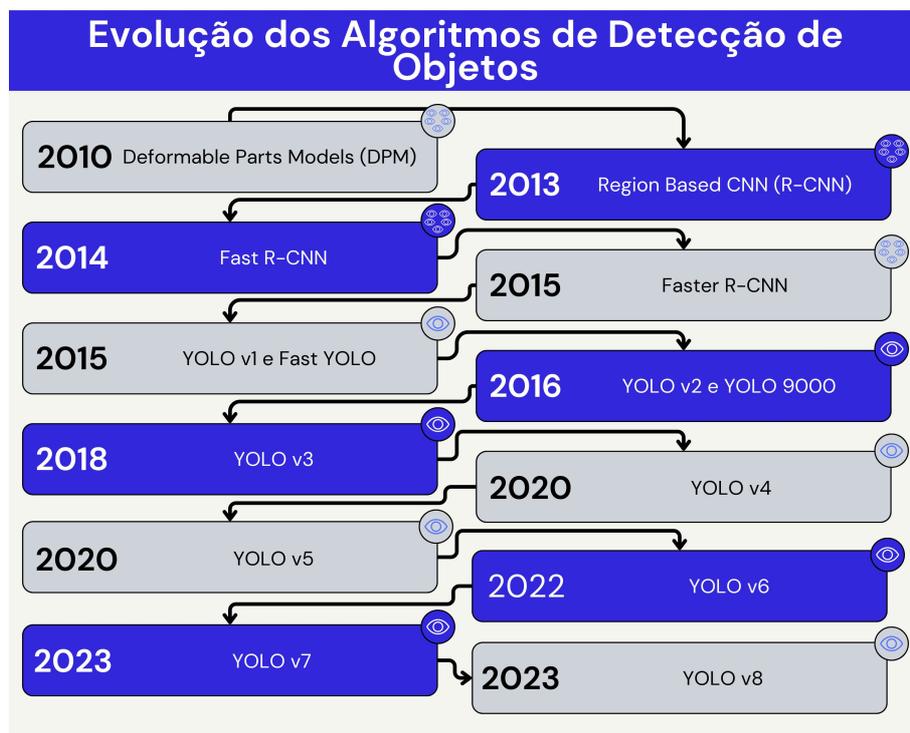
Devido ao grande sucesso alcançado pelos resultados da YOLO, os autores — já reconhecidos por sua ampla experiência na área de detecção de objetos — desenvolveram versões subsequentes significativamente aprimoradas, que foram posteriormente disponibilizadas à comunidade acadêmica. É relevante destacar que esses mesmos autores também contribuíram de forma expressiva para metodologias pregressas de destaque, como DPM, Fast R-CNN e Faster R-CNN (abordadas na Seção 3.1.1), o que reforça a solidez e a continuidade de sua atuação na área. Assim, o êxito da metodologia YOLO pode ser compreendido como fruto de uma trajetória consistente de pesquisa e inovação em detecção de objetos. A partir deste ponto, o termo "YOLO" será utilizado para referir-se especificamente ao trabalho seminal, salvo indicação em contrário, uma vez que existem diversas versões subsequentes. A Figura 6 apresenta uma visão geral da evolução das metodologias que compõem o estado da arte na detecção de objetos em imagens, destacando os melhores desempenhos obtidos até as respectivas datas de publicação.

As metodologias de dois estágios anteriores à YOLO foram discutidas na Seção 3.1.1 e, portanto, não serão retomadas nesta subseção. No ano seguinte ao lançamento da YOLO, os mesmos autores apresentaram duas versões subsequentes, denominadas YOLOv2 e YOLO9000 [57]. Já no título do artigo, os autores destacam que essas novas versões superam a YOLO original em termos de desempenho, velocidade e capacidade de generalização. A principal distinção entre as redes reside no processo de treinamento: a YOLO9000 foi pré-treinada para reconhecer mais de 9000 classes de objetos, ampliando significativamente sua aplicabilidade em cenários com grande diversidade de categorias.

Diversas modificações foram incorporadas à YOLOv2, contribuindo para caracterizar essa versão como superior à sua antecessora. As principais melhorias estão descritas a seguir:

- ***Batch Normalization***: A aplicação da técnica de *batch normalization* após todas as camadas convolucionais resultou em um aumento de 2% no valor de *mAP*, além de possibilitar a remoção dos mecanismos de *dropout* anteriormente utilizados, simplificando a arquitetura da rede;

Figura 6 – Evolução dos Algoritmos de Detecção de Objetos.



Fonte: Elaborado pelo Autor.

- **Aumento da Resolução das Imagens:** A resolução das imagens utilizadas no pré-treinamento foi elevada de  $224 \times 224$  para  $448 \times 448$ . Essa alteração exigiu que a rede aprendesse a extrair características em uma escala diferente daquela utilizada na fase inicial de treinamento. Para adaptar os filtros à nova resolução, parte do pré-treinamento foi realizada com a resolução aumentada durante algumas épocas. Essa estratégia contribuiu para um ganho de quase 4% no *mAP*;
- **Camadas Convolucionais com *Anchor Boxes*:** A predição das regiões de interesse, anteriormente realizada por meio de camadas totalmente conectadas, passou a ser baseada em caixas delimitadoras (*anchor boxes*) combinadas com deslocamentos (*offsets*). Essa mudança elevou o número potencial de regiões de interesse de 98 (na YOLO original) para milhares, conforme relatado pelos

autores. Embora essa alteração não tenha impactado diretamente o  $mAP$ , aumentou significativamente a capacidade de aprendizado da rede. Para otimizar a seleção das *anchor boxes*, foi aplicado o algoritmo de clusterização K-means às caixas delimitadoras do conjunto de treinamento, com o objetivo de identificar os centróides mais representativos. Determinou-se que a configuração ideal correspondia a cinco caixas por célula;

- **Aumento da Subdivisão da Imagem:** A grade de subdivisão da imagem foi expandida de  $7 \times 7$  para  $13 \times 13$ , com o intuito de melhorar a detecção de objetos pequenos. Essa modificação resultou em um acréscimo aproximado de 1% no valor do  $mAP$ ;
- **Treinamento em Multi-Resolução:** Foi adotada uma estratégia de variação dinâmica da resolução de entrada durante o treinamento, na qual a dimensão da imagem é alterada aleatoriamente a cada 10 *batches*, em múltiplos de 32 — valor escolhido em função dos *downsamplings* sucessivos da rede. Essa abordagem visa aumentar a robustez da rede na detecção de objetos em diferentes escalas e contextos.

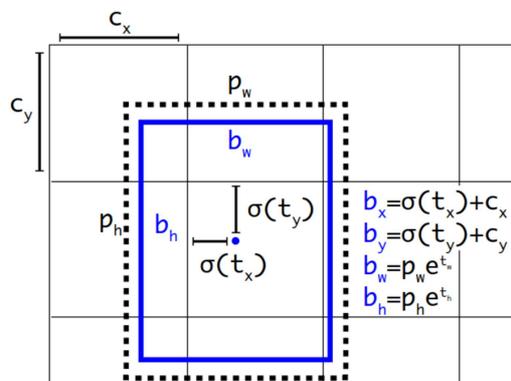
Além das melhorias supracitadas, outra importante inovação da YOLOv2 reside na reformulação da parametrização das *bounding boxes*, que supera limitações significativas da versão original. Enquanto a YOLO clássica estimava coordenadas absolutas arbitrárias na imagem - abordagem que frequentemente levava a instabilidades durante o treinamento e convergência lenta -, a YOLOv2 introduz um esquema de predição relativo às células da grade. Cada célula, localizada em  $(c_x, c_y)$  a partir do canto superior esquerdo da imagem, prediz cinco parâmetros transformados:  $(t_x, t_y, t_w, t_h, t_o)$ , onde  $t_o$  representa a *objectness score*.

A transformação para as coordenadas finais da caixa delimitadora é dada pela Equação 3.8.

$$\begin{aligned}
 b_x &= \sigma(t_x) + c_x \\
 b_y &= \sigma(t_y) + c_y \\
 b_w &= p_w e^{t_w} \\
 b_h &= p_h e^{t_h}
 \end{aligned}
 \tag{3.8}$$

Nesta formulação, a função sigmoide  $\sigma(t)$  garante que os deslocamentos  $(t_x, t_y)$  permaneçam no intervalo  $(0, 1)$ , ancorando assim a predição à célula de referência. Os termos exponenciais para  $(t_w, t_h)$  permitem um redimensionamento não-linear das dimensões das *anchor boxes*  $(p_w, p_h)$ , obtidas através de *k-means* sobre o conjunto de treinamento. Esta abordagem, ilustrada na Figura 7, oferece três vantagens fundamentais: (1) estabilização do treinamento através da restrição espacial das predições; (2) incorporação de conhecimento prévio via *anchors*; e (3) preservação da capacidade de prever boxes fora da proporção das *anchors* através dos parâmetros de transformação.

Figura 7 – Detecção das Caixas Delimitadoras.



Fonte: Retirado de [57].

Além dos parâmetros responsáveis por descrever o formato da caixa delimitadora, a rede também prediz a probabilidade de que a célula contenha um objeto, denotada por  $P_r(\text{object})$ , tal como na versão original da YOLO. A confiança da predição é então calculada pelo produto entre essa probabilidade e a sobreposição (IoU) entre a caixa predita ( $b$ ) e o objeto real, conforme mostra a Equação 3.9:

$$P_r(\text{object}) \times \text{IoU}(b, \text{object}) = \sigma(t_o) \quad (3.9)$$

Embora o conceito de confiança tenha sido mantido, a YOLOv2 introduziu uma parametrização aprimorada: a rede agora aprende um valor  $t_o$ , cuja saída é normalizada por uma função sigmoide, restringindo a confiança ao intervalo  $(0, 1)$ .

Essa mudança contribuiu para maior estabilidade no treinamento e resultou em um aumento de aproximadamente 5% no *mean Average Precision* (mAP) ao ser avaliado no banco de dados MS COCO.

Além da significativa melhoria no mAP, a YOLOv2 trouxe avanços notáveis em eficiência computacional, reduzindo substancialmente o tempo necessário para detecção de objetos. Esta otimização foi alcançada através de uma reformulação arquitetural profunda, resultando na *Darknet-19* - uma estrutura com 19 camadas convolucionais e 5 camadas de *max-pooling* cuidadosamente projetada para equilibrar precisão e velocidade.

Conforme detalhado em [57], a nova arquitetura alcançou um mAP de 78.6% no conjunto de dados PASCAL VOC 2007, com uma impressionante taxa de processamento de 40 frames por segundo em uma única GPU NVIDIA Titan X. Este desempenho supera não apenas a versão anterior da YOLO, mas também métodos contemporâneos como o *Faster R-CNN*, conforme evidenciado pelos resultados comparativos apresentados na Tabela 1. A combinação entre alta precisão e velocidade de inferência em tempo real posicionou a YOLOv2 como um marco no desenvolvimento de sistemas de detecção eficientes.

Como contribuição final, os autores introduziram uma inovadora estrutura hierárquica para organização das classes, denominada *WordTree*. Este modelo revolucionário organiza as classes semanticamente relacionadas em uma árvore de categorias, onde nós superiores representam conceitos mais gerais (como "animais") e folhas correspondem a classes específicas (como "golden retriever" ou "siamês").

A abordagem do *WordTree* resolveu dois desafios críticos: primeiro, permitiu a integração sinérgica de bancos de dados heterogêneos (ImageNet [55] e COCO [58]) através de um esquema unificado de classificação; segundo, possibilitou o treinamento em larga escala com mais de 9.000 classes distintas. A estrutura hierárquica melhora a generalização do modelo ao propagar informações entre classes relacionadas, enquanto mantém a capacidade de fazer previsões precisas em nível de subcategoria. Este avanço foi fundamental para a criação da YOLO9000, que estabeleceu novos paradigmas na detecção de objetos em grande escala.

Em 2018, Redmon e Farhadi apresentaram a YOLOv3 [59], uma evolução incremental que consolidou a metodologia como referência em detecção de obje-

tos em tempo real. A terceira iteração trouxe avanços significativos através da arquitetura *Darknet-53*, composta por 53 camadas convolucionais projetadas para extração hierárquica de características. Outra melhoria crucial foi a implementação de predição em múltiplas escalas, permitindo a análise simultânea em três diferentes resoluções de *feature maps* para capturar objetos de diversos tamanhos. Os autores também substituíram a função de ativação linear por uma função *logistic* para estimar as probabilidades de classe, resultando em predições mais robustas.

Apesar desses aprimoramentos técnicos - que incluíram melhor detecção de objetos pequenos e um balanceamento mais eficiente entre precisão e velocidade - os autores expressaram sérias preocupações éticas quanto às possíveis aplicações militares da tecnologia. Esta reflexão responsável levou ao encerramento de suas contribuições diretas no campo, embora o legado da YOLO continue a influenciar profundamente a pesquisa em visão computacional. A YOLOv3 permanece como outro marco fundamental, demonstrando como otimizações arquiteturais estratégicas podem elevar o desempenho sem comprometer a eficiência computacional que é marca registrada da abordagem YOLO.

Devido ao extraordinário sucesso da metodologia YOLO, a versão YOLOv4 [60] emergiu em 2020 como a primeira contribuição significativa desenvolvida por pesquisadores independentes da equipe original. Esta nova iteração introduziu aprimoramentos substanciais através de uma abordagem sistemática para seleção e combinação de técnicas *state-of-the-art*, organizadas em duas categorias principais: o *bag of freebies*, contendo otimizações que melhoram o processo de treinamento sem custo computacional adicional durante a inferência; e o *bag of specials*, composto por módulos especializados que aumentam a precisão da detecção à custa de um leve incremento na complexidade computacional.

Entre as inovações mais impactantes destaca-se a transição da arquitetura Darknet-53 para a CSPDarknet-53 [61], que incorpora o princípio de *Cross Stage Partial Networks* para reduzir redundâncias computacionais e melhorar o fluxo de gradientes durante o treinamento. Quando avaliada no banco de dados MS COCO, esta modificação permitiu alcançar 65 frames por segundo com precisão competitiva, estabelecendo novo padrão para sistemas de detecção em tempo real.

Os autores de YOLOv4 propuseram ainda uma estrutura conceitual ino-

vadora, decompondo a arquitetura de detecção em três componentes funcionais distintos:

- *Backbone*: Rede base (CSPDarknet-53) responsável pela extração hierárquica de características;
- *Neck*: Camadas intermediárias (SPP, PANet) que agregam e refinam características multi-escala;
- *Head*: Módulo final que realiza a classificação e regressão das bounding boxes.

Esta decomposição modular não apenas facilitou a compreensão e análise das arquiteturas de detecção, mas também estabeleceu um paradigma que influenciaria profundamente o desenvolvimento de versões subsequentes. A combinação estratégica destes componentes, juntamente com técnicas avançadas de aumento de dados e otimização, permitiu que a YOLOv4 superasse significativamente suas predecessoras tanto em precisão quanto em eficiência computacional.

Em 2020, a YOLOv5 [62], desenvolvida pela Ultralytics, marcou uma evolução significativa na família de arquiteturas YOLO, alcançando desempenho competitivo no benchmark MS COCO. Diferentemente das versões anteriores, a YOLOv5 introduziu duas inovações fundamentais: a transição da estrutura Darknet (implementada em C) para o framework PyTorch, e a disponibilização de modelos pré-treinados em cinco escalas distintas (*nano*, *small*, *medium*, *large* e *x-large*). Esta mudança de implementação trouxe maior acessibilidade à comunidade de pesquisa, facilitando tanto a compreensão quanto a adaptação do código para diferentes aplicações.

Embora a YOLOv5 não tenha sido acompanhada por uma publicação científica formal revisada por pares - fato que gerou debates na comunidade - seu desempenho foi empiricamente validado através de diversos estudos independentes [63, 64, 65]. As cinco variantes da arquitetura oferecem um *trade-off* controlado entre precisão e eficiência computacional, permitindo que usuários selecionem o modelo mais adequado às suas restrições operacionais. A versão *nano*, por exemplo, é otimizada para dispositivos com recursos limitados, enquanto a *x-large* prioriza máxima acurácia.

Adicionalmente, a YOLOv5 introduziu capacidades de segmentação de imagens sem modificações estruturais significativas, expandindo assim seu escopo de aplicação. Desde seu lançamento inicial, a arquitetura passou por contínuos refinamentos, atingindo atualmente a versão 5.7, que incorpora melhorias incrementais tanto em precisão quanto em eficiência computacional.

Após dois anos, foi lançada a versão YOLOv6 pela empresa Meituan, cujos detalhes podem ser consultados no relatório técnico [66]. Em 2023, a metodologia recebeu uma atualização significativa, resultando na versão 6.3 da YOLO [67], desenvolvida com o objetivo de aprimorar ainda mais os resultados de acurácia. Os autores promoveram uma reformulação abrangente no *backbone* da rede, investigando diferentes variações de camadas convolucionais simples e empilhadas, bem como a utilização de distintas funções de ativação ao longo da arquitetura. Além disso, foram implementadas melhorias adicionais voltadas ao aumento do desempenho na tarefa de classificação. A YOLOv6 foi disponibilizada em cinco versões de tamanhos distintos, sendo que a menor delas é capaz de atingir até 1234 FPS no processo de detecção.

Em 2023, os mesmos autores responsáveis pela YOLOv4 apresentaram a versão 7 da metodologia [68]. Assim como na versão anterior, foram implementadas diversas melhorias conhecidas como *bag-of-freebies*, com o objetivo de otimizar o processo de treinamento da rede. Tais aprimoramentos possibilitaram uma expressiva redução na quantidade de parâmetros do modelo. Em comparação com a YOLOv4, a YOLOv7 apresentou uma diminuição de 75% no número de parâmetros, o que resultou em uma redução de aproximadamente 36% no custo computacional. Além disso, os autores disponibilizaram duas versões pré-treinadas da rede, diferenciadas principalmente pelo tamanho, seguindo a mesma abordagem adotada nas versões 5 e 6 da YOLO.

Além das versões numeradas discutidas neste trabalho, outras variantes da YOLO com grande relevância também foram desenvolvidas ao longo dos anos. Em 2021, foi apresentada a YOLOX [69], cujo principal objetivo era aumentar a eficiência na detecção de objetos, utilizando os *backbones* Darknet-53 e CSPNet [61]. No entanto, os autores propuseram uma abordagem que se distancia da estrutura tradicional baseada na separação entre *backbone*, *neck* e *head*. Essa reformulação

arquitetural resultou em um modelo com convergência mais rápida em relação às versões anteriores da YOLO, concentrando o processo de detecção na etapa de *head*.

### 3.1.6 Detecção de Objetos - YOLOv8

A YOLOv8 é uma arquitetura avançada para detecção de objetos que demonstrou desempenho superior em comparação com outras metodologias da família YOLO disponíveis à época de seu lançamento, consolidando-se como uma referência nas pesquisas contemporâneas em visão computacional [70]. Desenvolvida pela Ultralytics como sucessora direta da YOLOv5, essa versão foi avaliada utilizando o conjunto de dados MS COCO [71], seguindo os mesmos protocolos experimentais adotados nas versões anteriores, de modo a permitir uma comparação direta de desempenho.

Além de aprimorar os mecanismos de detecção de objetos tradicionais, a YOLOv8 introduz um *framework* unificado que incorpora múltiplas funcionalidades, tais como segmentação semântica, estimativa de pose, rastreamento de múltiplos objetos e classificação de imagens. Embora ainda não tenha sido formalmente publicada em periódicos científicos com revisão por pares, sua eficácia tem sido amplamente validada na literatura especializada, especialmente em aplicações envolvendo veículos aéreos não tripulados (VANTs) [72, 73]. A arquitetura geral da rede é ilustrada na Figura 8.

O *backbone* da YOLOv8, ilustrado na Figura 8, constitui o componente fundamental para extração hierárquica de características espaciais que definem os limites dos objetos em cenas complexas. Esta arquitetura é organizada em cinco estágios progressivos de processamento, cada um especializado na captura de padrões em diferentes níveis de abstração. O primeiro estágio, denominado *ConvModule*, atua como camada de entrada e é composto por:

$$\text{ConvModule} = \text{Conv2d}(k = 3 \times 3) \circ \text{BatchNorm} \circ \text{SiLU} \quad (3.10)$$

onde  $\circ$  denota composição de operações, e a função SiLU (*Sigmoid Linear Unit*), definida como  $\text{SiLU}(x) = x \cdot \sigma(x)$ , oferece vantagens significativas sobre a



- Preserva a precisão na detecção de bordas.

Esta combinação de componentes - *ConvModule*, C2F e SPPF - forma uma arquitetura coesa que equilibra eficiência computacional com capacidade discriminativa, permitindo à YOLOv8 alcançar desempenho superior em tarefas de detecção de objetos.

Os estágios 3, 4 e 5 do *backbone* estão conectados à estrutura *neck*, que funciona como uma camada intermediária crítica para o processamento e fusão de características multi-escala. Este módulo emprega uma arquitetura bifuncional que combina:

$$\text{Neck} = \mathcal{F}_{\text{C2F}} \circ \mathcal{U}_{\text{upsample}} \oplus \mathcal{C}_{\text{concat}} \quad (3.11)$$

onde  $\mathcal{F}_{\text{C2F}}$  representa os módulos Coarse-to-Fine,  $\mathcal{U}_{\text{upsample}}$  as operações de upsample bilinear, e  $\mathcal{C}_{\text{concat}}$  as concatenações estratégicas de *features*. Esta combinação permite:

- Fusão eficiente de características em diferentes níveis de abstração;
- Preservação de informações espaciais para objetos pequenos;
- Propagação de contexto semântico através das escalas.

A componente *head* da YOLOv8 implementa um mecanismo de detecção tripartido que opera simultaneamente em três resoluções espaciais distintas. Cada ramo de detecção é composto por:

$$\text{Head} = \text{ConvModule}^{\times 3} \circ \text{Conv2d}(k = 1 \times 1) \quad (3.12)$$

Esta configuração permite a detecção robusta de objetos em múltiplas escalas, desde instâncias pequenas até objetos grandes que ocupam porções significativas da imagem. Vale destacar que a arquitetura mantém consistência estrutural em todas as variantes (YOLOv8n-x), com as diferenças de desempenho sendo determinadas exclusivamente pela largura da rede (número de canais), profundidade (número de camadas) e tamanho dos *kernels* convolucionais.

A Figura 8 ilustra este fluxo completo de processamento, destacando como a integração entre *backbone*, *neck* e *head* forma uma arquitetura coerente para detecção de objetos em tempo real.

### 3.2 MAPAS 3D UTILIZANDO A ESTRUTURA OCTOMAP

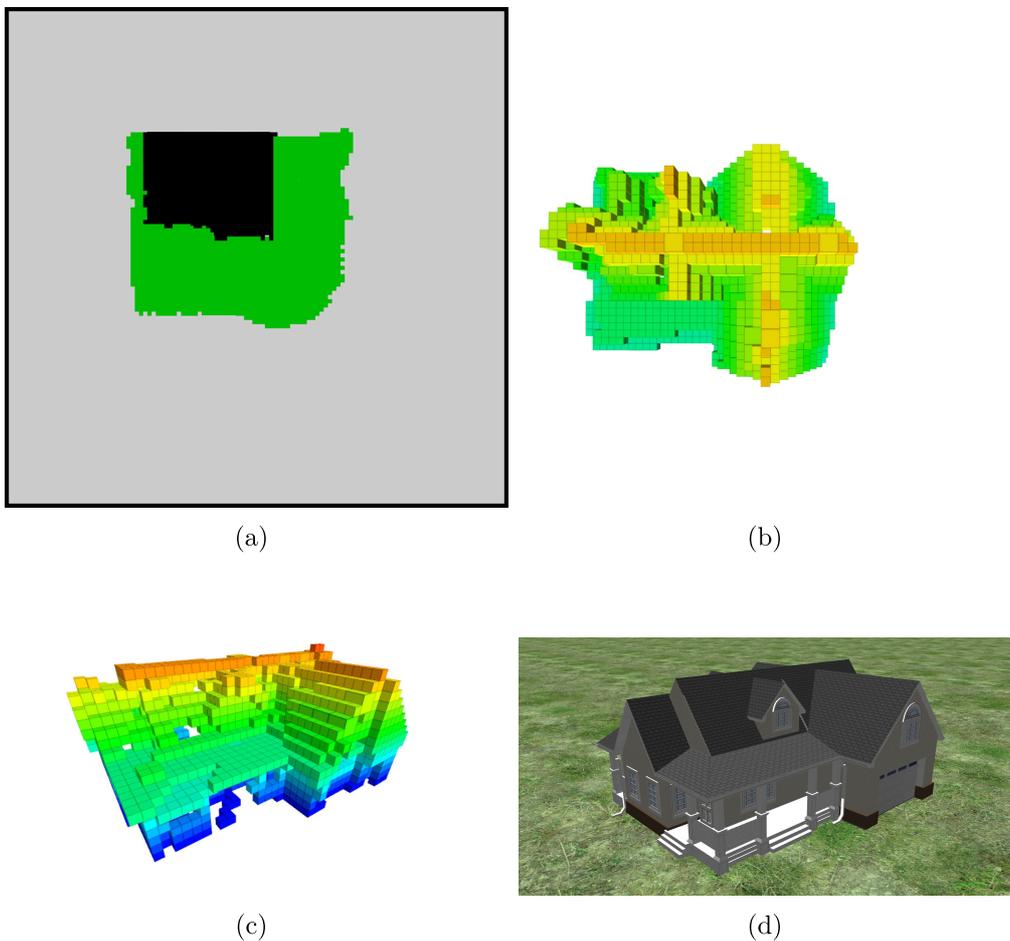
A representação espacial em sistemas robóticos de exploração pode ser categorizada em três paradigmas fundamentais, cada um com características distintas. Os mapas 2D oferecem uma representação planar eficiente para ambientes estruturados com navegação em um único nível, sendo computacionalmente leves e de implementação direta. No entanto, sua limitação dimensional os torna inadequados para plataformas aéreas ou ambientes com múltiplos níveis, conforme demonstrado por [22].

A abordagem 2.5D, introduzida por [75], estende o modelo 2D ao incorporar informações de elevação por célula. Esta representação melhora a modelagem de terrenos irregulares, mas permanece incapaz de capturar estruturas complexas como pontes sobre vias ou edifícios com múltiplos andares. Como observado em [22], a natureza implícita da dimensionalidade vertical nestes mapas leva à perda de informação espacial crítica para navegação tridimensional.

Os mapas 3D emergem como solução abrangente para representação volumétrica, superando as limitações das abordagens anteriores. Embora demandem maior poder computacional [76], eles permitem modelagem precisa de ambientes complexos e navegação sem restrições em todas as dimensões espaciais. Para a aplicação proposta neste trabalho - navegação autônoma de drones em espaço livre - a adoção de mapas 3D é indispensável, pois acomoda variações dinâmicas de altitude e estruturas verticais complexas. A Figura 9 apresenta uma comparação visual entre estas três abordagens, destacando suas capacidades representativas.

A Figura 9 ilustra comparativamente as três abordagens de representação espacial discutidas. Na configuração 2D (Figura 9(a)), os obstáculos são codificados em preto, formando uma projeção planar que não distingue variações verticais. Esta limitação torna impossível identificar áreas navegáveis sob estruturas suspensas, como varandas ou pontes, restringindo a movimentação do robô ao plano  $XY$ .

Figura 9 – (a) Mapa 2D (b) Mapa 2.5D (c) Mapa 3D (d) Objeto Mapeado.



Fonte: Elaborado pelo Autor.

Tal representação é adequada apenas para aplicações onde a navegação ocorre exclusivamente em um único nível horizontal.

A abordagem 2.5D (Figura 9(b)) introduz um eixo vertical discreto, permitindo a representação de elevações distintas em cada célula do mapa. Embora esta extensão permita ao robô navegar em diferentes níveis de altitude - aumentando assim a área explorável em comparação com mapas 2D puros - ela ainda falha em capturar adequadamente estruturas sobrepostas no mesmo plano horizontal. O caso da varanda exemplifica esta limitação: enquanto o mapa 2.5D pode representar sua altura, não consegue expressar simultaneamente o espaço navegável abaixo dela.

A solução definitiva é oferecida pelo mapeamento 3D completo (Figura 9(c)), que preserva explicitamente a informação volumétrica do ambiente. Nesta representação, tanto a varanda quanto o espaço abaixo dela são corretamente modelados como regiões distintas, permitindo ao robô identificar todas as áreas potencialmente navegáveis. Esta fidelidade espacial é essencial para sistemas que exigem navegação verdadeiramente tridimensional, particularmente em ambientes complexos com múltiplos níveis interconectados.

A representação 3D do ambiente pode ser discretizada em células volumétricas (*voxels*) [77], cuja resolução espacial impacta diretamente na fidelidade da representação. Contrariamente à intuição inicial, resoluções menores (com *voxels* maiores) frequentemente proporcionam melhor eficiência computacional sem comprometer significativamente a utilidade do mapa para navegação. Cada *voxel* deve codificar três estados possíveis: livre, ocupado ou desconhecido, sendo este último essencial para áreas não exploradas. A complexidade computacional inerente a esta abordagem levou ao desenvolvimento de técnicas alternativas, como nuvens de pontos, que porém herdam as limitações de precisão dos sensores que as geram [22].

Para mitigar o alto custo computacional do mapeamento 3D, estruturas hierárquicas como *octrees* foram adaptadas desde seu conceito original [78] para aplicações robóticas. Trabalhos subsequentes [79, 80] introduziram atualizações probabilísticas, porém sem resolver completamente os desafios de gestão de memória em ambientes dinâmicos e de grande escala.

Neste trabalho, adotamos o *framework* OctoMap [22], uma implementação em C++ que resolve quatro requisitos fundamentais para mapeamento 3D eficaz:

- Representação Completa: Codificação explícita dos três estados espaciais (livre, ocupado, desconhecido) em estrutura volumétrica verdadeiramente tridimensional, essencial para navegação em ambientes não estruturados;
- Atualização Probabilística: Mecanismo de atualização Bayesiana que incorpora incertezas sensoriais, permitindo fusão robusta de múltiplas observações ao longo do tempo;
- Flexibilidade Multi-resolução: Capacidade de adaptar dinamicamente a granularidade da representação, possibilitando planejamento global (alta resolução)

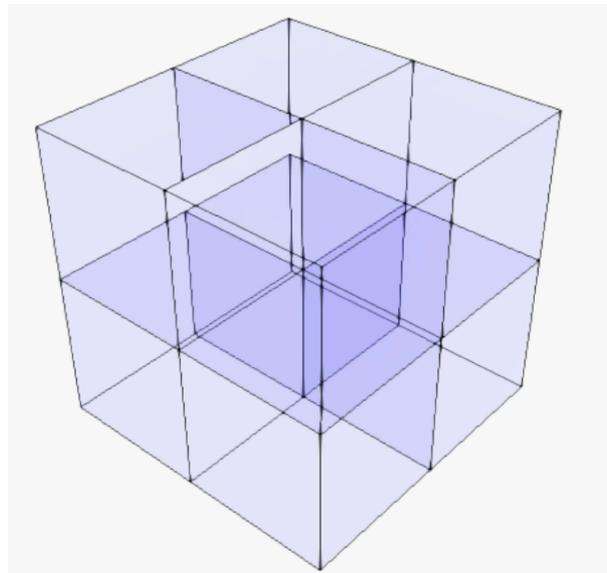
e local (baixa resolução) na mesma estrutura de dados;

- Otimização de Memória: Uso eficiente de recursos através de representação hierárquica (*octree*), compactação de regiões homogêneas, e alocação sob demanda, viabilizando aplicações em larga escala.

Esta combinação de características faz do OctoMap uma solução particularmente adequada para exploração autônoma, equilibrando fidelidade espacial com eficiência computacional. A estrutura *octree* subjacente permite ainda operações eficientes como consultas de colisão e integração de novas observações, fundamentais para sistemas de navegação em tempo real.

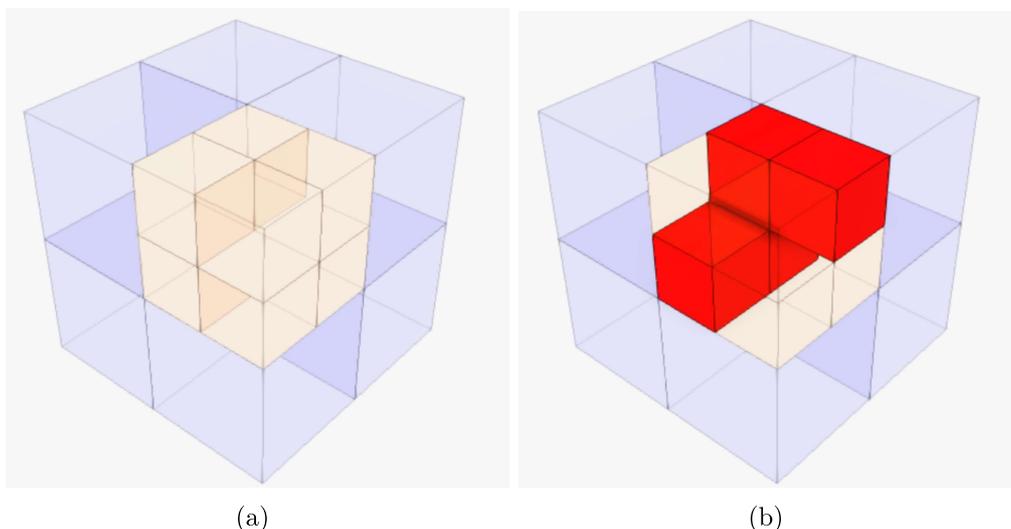
O conceito de *octree* foi introduzido inicialmente em [78], com o principal objetivo de organizar informações de maneira hierárquica através de árvores. Cada nó na árvore (*tree*) gera outras 8 ramificações possíveis (octetos). Aplicando esta representação ao ambiente 3D, o espaço é subdividido em cubos de tamanho uniforme para representar o ambiente. A Figura 10 ilustra essa subdivisão do ambiente.

Figura 10 – Modelo Inicial da *Octree*.



Fonte: Elaborado pelo Autor.

Figura 11 – (a) Subdivisão em Novos 8 Octetos (b) Atualização dos Novos Estados dos Octetos.



Fonte: Elaborado pelo Autor.

Considerando que todo o ambiente é contido em apenas 8 subdivisões no modelo inicial, fica evidente a eficiência da *octree* na representação de um mapa 3D. Cada célula dos 8 octetos pode ser subdividida em outras 8 células e assim por diante. Essa divisão é possível até um limite pré-definido, cuja granularidade impactará diretamente na representação detalhada do ambiente. Cada octeto sempre será caracterizado por um dos três estados mencionados anteriormente: obstáculo, livre ou desconhecido.

Quando uma região dentro de um octeto apresenta um estado diferente daquele definido, é realizada uma nova subdivisão (ver Figura 11 (a)). As 8 novas células geradas pela subdivisão serão reclassificadas de acordo com os novos estados resultantes das observações adicionais realizadas pelos sensores. A Figura 11 (b) representa em vermelho três novas regiões.

O OctoMap constitui uma metodologia sofisticada para gestão de mapas 3D baseada em *octrees*, caracterizando-se como um sistema de multi-resolução devido à sua estrutura hierárquica adaptativa. O cerne desta abordagem reside no modelo probabilístico que atribui a cada *voxel*  $n$  uma probabilidade  $P(n)$  de ocupação,

atualizada dinamicamente mediante novas observações sensoriais.

O processo de atualização segue uma formulação bayesiana expressa por:

$$P(n|z_{l:t}) = \left[ 1 + \frac{1 - P(n|z_t)}{P(n|z_t)} \frac{1 - P(n|z_{l:t-1})}{P(n|z_{l:t-1})} \frac{P(n)}{1 - P(n)} \right]^{-1} \quad (3.13)$$

Nesta formulação,  $P(n|z_{l:t})$  representa a probabilidade atualizada do *voxel*  $n$  estar ocupado considerando todas as observações desde o instante inicial até o tempo corrente  $t$ . A probabilidade a priori  $P(n)$  é tipicamente inicializada como 0.5, refletindo máxima incerteza inicial. O termo  $P(n|z_{l:t-1})$  incorpora o histórico de observações anteriores, enquanto  $P(n|z_t)$  considera apenas a medida mais recente.

Para otimização computacional, os autores converteram esta representação para o domínio logarítmico através da transformação *log-odds*:

$$L(n|z_{l:t}) = L(n|z_{l:t-1}) + L(n|z_t) \quad (3.14)$$

onde  $L(n)$  é definido como:

$$L(n) = \ln \left[ \frac{P'(n)}{1 - P'(n)} \right] \quad (3.15)$$

Esta transformação oferece vantagens computacionais significativas: elimina operações não-lineares complexas, melhora a estabilidade numérica para probabilidades extremas (próximas de 0 ou 1), e permite atualizações eficientes através de simples adições. A abordagem probabilística do OctoMap, aliada à sua estrutura hierárquica, viabiliza a manutenção eficiente de mapas consistentes mesmo em ambientes dinâmicos e de grande escala, demonstrando robustez frente a incertezas sensoriais.

Diante de uma nova observação, cada célula do mapa pode assumir apenas duas possibilidades: presença de obstáculo ( $P'(n)$ ) ou ausência de obstáculo ( $1 - P'(n)$ ). Com essa notação, é possível reescrever a Equação 3.13, que envolve multiplicações sucessivas, em uma forma aditiva utilizando log-odds, conforme representado na Equação 3.14.

Para que uma célula seja considerada ocupada, a probabilidade de ocupação  $P(n|z_l : t)$  deve ultrapassar um valor limiar. Caso contrário, a célula é classificada como livre, ou seja, sem obstáculos. Analisando a Equação 3.14, observa-se que, se uma célula foi marcada como livre em  $k$  observações consecutivas, será necessário pelo menos o mesmo número de observações positivas (com obstáculo) para que o estado da célula seja revertido para ocupado. Essa definição é adequada para ambientes estáticos, nos quais a presença ou ausência de obstáculos tende a permanecer constante.

Entretanto, em cenários dinâmicos — como os abordados neste trabalho — essa característica torna-se indesejável, pois introduz um atraso na atualização do mapa. Para mitigar esse problema, foi proposta em [81] uma modificação na política de atualização dos voxels. Os autores sugerem restringir os valores mínimo e máximo da variável  $L(n)$ , limitando seu crescimento, o que permite atualizações mais responsivas e robustas em ambientes dinâmicos. Com isso, a Equação 3.14 é reformulada conforme apresentado na Equação 3.16.

$$L(n|z_{l:t}) = \max(\min(L(n|z_{l:t-1}) + L(n|z_t), l_{\max}), l_{\min}), \quad (3.16)$$

Os valores mínimo e máximo de atualização são representados por  $l_{\min}$  e  $l_{\max}$ , respectivamente. Além de evitar que o valor de  $L(n)$  diverja com o tempo, essa modificação garante uma rápida adaptação do estado da célula diante de mudanças nas observações. Essa característica é especialmente relevante em ambientes dinâmicos, onde a atualização ágil das informações é essencial.

Adicionalmente, o valor da probabilidade associada a cada célula desempenha um papel fundamental na tarefa de *pruning*, que consiste na remoção de células desnecessárias do mapa. Esse processo visa otimizar a eficiência computacional, reduzindo a complexidade da estrutura mantida em memória sem comprometer a representação do ambiente.

Outra característica distintiva do OctoMap é sua capacidade de consulta em múltiplas resoluções, possibilitada por sua estrutura hierárquica baseada em octrees. Essa propriedade permite utilizar, conforme a necessidade, informações provenientes de diferentes níveis de detalhe do mapa — resoluções mais finas (nós

folha) ou mais grosseiras (nós internos).

Uma estratégia eficiente para inferir o estado de ocupação  $l(n)$  de um nó interno foi proposta em [82]. Nessa abordagem, o estado do nó é calculado como a média dos valores de ocupação dos seus oito filhos (octetos), conforme definido na Equação 3.17.

$$\bar{l}(n) = \frac{1}{8} \sum_{i=1}^8 L(n_i) \quad (3.17)$$

Outra estratégia interessante, embora conservadora, é considerar o valor máximo dos octetos para representar o estado do nó  $n$ . Essa abordagem pode ser observada na Equação 3.18 e é particularmente útil ao utilizar um planejador de caminhos baseado em um OctoMap para evitar colisões.

$$\bar{l}(n) = \max_i(L(n_i)) \quad (3.18)$$

Para atualização eficiente das células, os autores empregaram uma adaptação tridimensional do algoritmo de Bresenham [83], otimizado para operações em *voxels*. Durante cada iteração de atualização, o sensor é virtualmente projetado no ambiente, classificando os feixes laser em duas categorias distintas:

$$L(n|z_t) = \begin{cases} l_{\text{free}} & \text{se o feixe atravessou o } \textit{voxel} \text{ sem detecção} \\ l_{\text{occ}} & \text{se o feixe detectou obstáculo no } \textit{voxel} \end{cases} \quad (3.19)$$

onde  $L(n|z_t)$  representa a atualização no domínio logarítmico para o *voxel*  $n$  na observação  $z_t$ . Através de extensa validação experimental, os autores determinaram os valores ótimos  $l_{\text{occ}} = 0.85$  e  $l_{\text{free}} = -0.4$ , que correspondem, respectivamente, às probabilidades de ocupação:

$$P_{\text{occ}} = \frac{e^{l_{\text{occ}}}}{1 + e^{l_{\text{occ}}}} = 0.7 \quad \text{e} \quad P_{\text{free}} = \frac{e^{l_{\text{free}}}}{1 + e^{l_{\text{free}}}} = 0.4 \quad (3.20)$$

Para garantir estabilidade numérica, foram estabelecidos limites de saturação  $l_{\text{min}} = -2$  e  $l_{\text{max}} = 3.5$ , correspondentes às probabilidades limites de  $P_{\text{min}} = 0.12$

e  $P_{\max} = 0.97$ . Estes limites previnem a convergência prematura para certezas absolutas ( $P = 0$  ou  $P = 1$ ), mantendo a capacidade de adaptação a mudanças no ambiente.

A metodologia OctoMap, com seu mecanismo eficiente de atualização, apresenta uma solução elegante para o mapeamento 3D em ambientes dinâmicos, sendo especialmente adequada para aplicações de exploração autônoma, como as propostas neste trabalho. Sua robustez decorre da combinação de três aspectos fundamentais: projeção eficiente de raios via algoritmo de Bresenham 3D, atualização probabilística com valores limitados que garantem rápida adaptação às mudanças, e uma estrutura hierárquica baseada em *octree*, que possibilita consultas em múltiplas resoluções e otimização da memória através do *pruning* de células desnecessárias.

### 3.3 PLANEJAMENTO DE CAMINHOS

Uma das atividades indispensáveis em operações de exploração autônoma é a resolução do problema de planejamento de caminhos. A partir da posição atual do robô, é crucial encontrar um caminho viável e desobstruído para que ele possa alcançar o destino desejado. Assim, o planejador de caminhos fornece ao robô um conjunto ordenado de pontos (*waypoints*), cuja execução sequencial conduzirá o robô à posição final desejada. Este problema é relevante tanto em ambientes bidimensionais quanto tridimensionais, alinhando-se aos objetivos deste trabalho.

Os algoritmos para este tipo de variação podem ser divididos em 4 categorias conforme classificação apresentada em [84]:

- **Algoritmos Baseados em Amostragem:** Técnicas probabilísticas que constroem trajetórias através de amostragem aleatória ou dirigida no espaço de configuração. Destacam-se o *Probabilistic Road Maps* (PRM) [85], que constrói um grafo de conectividade, e a família de algoritmos derivados do Rapidly exploring random tree [31], que expandem uma árvore de forma incremental. Embora eficientes em espaços de alta dimensionalidade, não garantem optimalidade geométrica do caminho gerado.
- **Algoritmos Ótimos Baseados em Nó:** Métodos de busca sistemática que expandem nós a partir da configuração inicial, garantindo optimalidade sob

certas condições. Os principais representantes são as variações do A\* [86] e o algoritmo de Dijkstra [87]. Sua eficácia está diretamente vinculada à qualidade da heurística adotada, e o custo computacional pode se tornar proibitivo em espaços complexos.

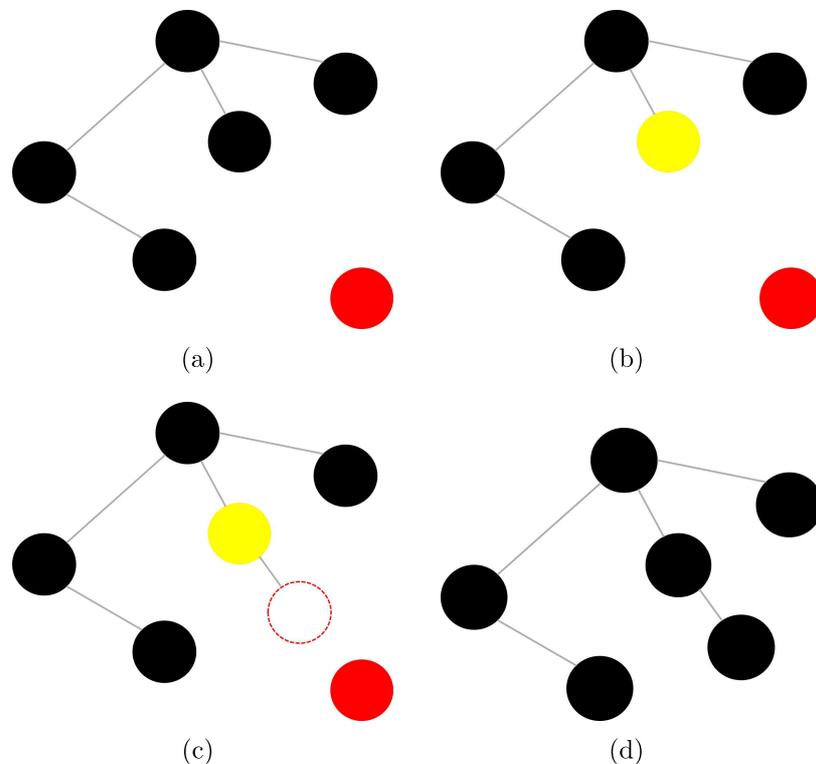
- **Algoritmos Bio-Inspirados:** Técnicas inspiradas em processos naturais, incluindo algoritmos evolutivos (como otimização por enxame de partículas [88], colônias de formigas [89] e algoritmos genéticos [90]) e abordagens baseadas em redes neurais. Embora versáteis, exigem cuidadoso ajuste de parâmetros e podem apresentar convergência lenta em espaços de busca complexos.
- **Algoritmos Híbridos:** Combinações sinérgicas de diferentes paradigmas, como o método P-BI-RRT proposto em [91] que integra PRM com RRT. Estas abordagens visam aproveitar as vantagens complementares de diferentes metodologias para superar suas respectivas limitações individuais.

Dada a contextualização e revisão bibliográfica realizadas, optou-se por escolher um método baseado no RRT devido ao problema de alta dimensionalidade da exploração autônoma a ser resolvido. Como evidência dessa escolha, observa-se que vários trabalhos recentes utilizam algum algoritmo inspirado no RRT original como apresentado em [92, 93, 94].

A metodologia seminal RRT foi desenvolvida em 1998 [31], e os detalhes para sua implementação foram apresentados em [95]. A ideia principal consiste em espalhar um conjunto de amostras pelo mapa de maneira sequencial e aleatória, daí o motivo do termo *random* (do português, aleatório) no nome da metodologia. O processo para a concepção do caminho inicia com um nó inicial na posição em que o robô se encontra.

A árvore que se expande a partir da raiz (nó inicial ou nó raiz) consiste na inclusão de novos nós a partir de amostras que são obtidas de maneira aleatória. Para cada amostragem realizada, é necessário buscar na árvore já existente o nó mais próximo da amostra candidata. A este nó, damos o nome de nó referência. Dessa forma, o candidato a ser um nó será o resultado da projeção do nó mais próximo (referência) em relação à amostra aleatória. Essa projeção é parametrizada por um coeficiente que define o tamanho dessa projeção. Se não houver obstáculos

Figura 12 – Expansão da Árvore Baseado no Método RRT: (a) Amostragem Aleatória. (b) Detecção do Nó Mais Próximo. (c) Projeção do Nó Mais Próximo até o Amostrado. (d) Incorporação do Nó Projetado à Árvore.



Fonte: Elaborado pelo Autor.

nessa projeção, o ponto mais distante em relação ao nó de referência (final da projeção) é adicionado à árvore como um nó permanente. Essa adição consiste em definir a posição cartesiana do nó e uma relação de conexão entre o novo nó e o mais próximo dele (*edge*). Os passos para a expansão da árvore foram ilustrados na Figura 12.

A Figura 12(a) apresenta o nó amostrado em uma posição aleatória, representada por um círculo vermelho. A partir dessa amostragem, determina-se o nó mais próximo (Figura 12(b)), marcado pelo círculo amarelo. Em seguida, realiza-se uma projeção do nó mais próximo em direção à posição amostrada (Figura 12(c)), ilustrada como um círculo tracejado com bordas vermelhas. Se não houver colisão, o novo nó é incorporado à árvore representado através da Figura 12(d).

O critério de parada desta metodologia consiste em encontrar um nó próximo ao ponto objetivo para ser adicionado. Uma das grandes vantagens do RRT é sua capacidade de explorar rapidamente o espaço de trabalho, sem a necessidade de discretização do ambiente. Essa característica é particularmente útil em ambientes de alta dimensionalidade. Ao término da fase de expansão, realiza-se a busca inversa a partir do último nó adicionado à árvore, visando encontrar o caminho mais curto de volta ao ponto inicial através dos *edges* definidas durante a expansão. É crucial destacar que o preço pago pela rapidez na geração do caminho é a possibilidade de não se obter caminhos otimizados.

Os resultados obtidos com o RRT destacaram-se pela rapidez e eficiência na expansão da árvore em diferentes ambientes, o que estimulou o desenvolvimento de novas metodologias baseadas no RRT original. Os mesmos autores propuseram uma segunda abordagem, denominada RRT-Connect, em [96]. A ideia principal é criar duas árvores: uma partindo do ponto inicial e outra partindo do ponto objetivo. Além disso, uma heurística é definida para estender a projeção do nó de referência até alcançar a posição amostrada ou encontrar um obstáculo, eliminando assim a necessidade do parâmetro de projeção na metodologia original. O critério de parada é alcançado quando um nó em uma das árvores se conecta à outra. Essas modificações resultaram em uma redução no tempo necessário para encontrar um caminho em comparação ao RRT. No entanto, como discutido anteriormente, os caminhos encontrados podem não ser ótimos em termos de distância. Um resultado significativo demonstrado por este trabalho é a capacidade de expandir a árvore levando em consideração as restrições impostas pelo modelo cinemático do robô.

Para alcançar caminhos ótimos, surgiu a metodologia RRT\* [97], cujo nome é uma referência ao algoritmo de planejamento de caminho ótimo A\* [86]. A principal diferença em relação ao RRT é a incorporação de duas funções no processo de adição de novos nós. Primeiramente, é realizada uma análise dos custos de caminho de todos os nós que estão dentro de uma região próxima ao nó de referência. O objetivo dessa análise é conectar o novo nó à aresta que possui o menor custo para alcançar o nó raiz. Essa modificação foi crucial, pois o nó mais próximo à posição amostrada nem sempre apresentaria o menor custo para guiar até o ponto objetivo. Em segundo lugar, é feita uma análise de todos os nós

presentes em uma área próxima ao nó de referência. Após a inserção de um novo nó, verifica-se se é possível encontrar um caminho com custo menor realocando os nós nessa região. Ao contrário do RRT básico, o RRT\* é *asymptotically optimal*, ou seja, converge para a solução ótima quando o número de amostras tende ao infinito, mantendo a completude probabilística da abordagem original.

### 3.3.1 Variante Informed-RRT\*

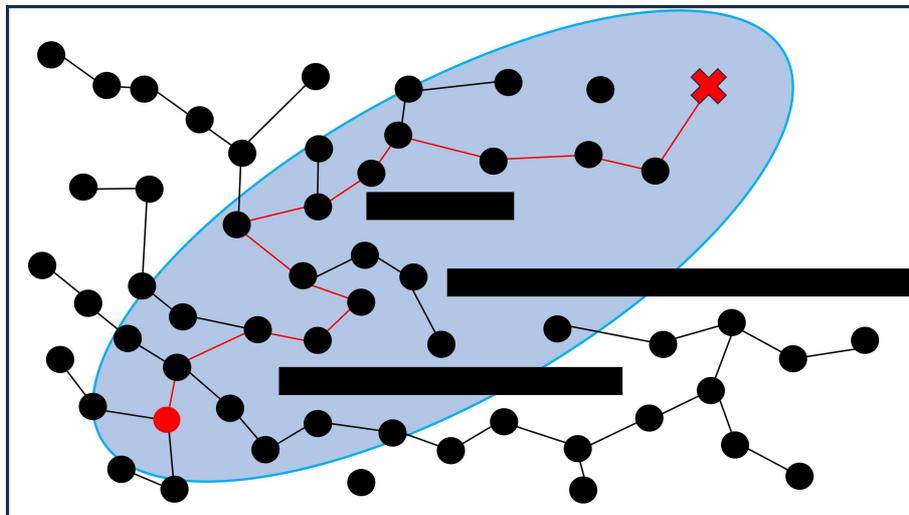
A técnica RRT\* trouxe uma significativa melhoria na eficiência da exploração de ambientes de alta dimensionalidade, alcançando resultados ótimos. No entanto, estudos subsequentes buscaram melhorar a velocidade na descoberta de caminhos, como o Informed-RRT\* [24]. Este método introduz uma heurística elipsoidal para guiar as amostras durante o processo de busca. Após obter um caminho inicial (que pode não ser ótimo), uma elipse é definida com focos posicionados na origem e no destino. O tamanho dessa elipse é determinado pelos custos do caminho mínimo ( $c_{min}$ ) e do caminho inicial encontrado ( $c_{max}$ ), conforme ilustrado na Figura 13. Essa estratégia garante que qualquer caminho com custo inferior estará contido dentro da elipse predefinida. Até que um caminho com custo menor que  $c_{max}$  seja encontrado, este pode ser considerado o melhor caminho viável. Por esse motivo, em alguns trabalhos  $c_{max}$  também pode ser referido como  $c_{best}$ .

A partir desse ponto, as amostras são concentradas exclusivamente na região interna da elipse. À medida que caminhos mais curtos são descobertos, a elipse é reajustada, reduzindo gradualmente sua área. Os autores demonstraram que o Informed-RRT\* preserva as propriedades de completude probabilística e *asymptotically optimal* observadas no RRT\*, garantindo assim uma abordagem confiável para o planejamento de trajetórias. A construção da elipse é mostrada pela Figura 14.

É importante destacar que  $c_{min}$  corresponde ao caminho ótimo, que é uma linha reta ligando a posição inicial ao objetivo. Esse caminho é simbólico e utilizado para gerar a elipse que orientará a heurística na busca de caminhos, porém pode ser impossível de ser atingido no cenário de exploração real.

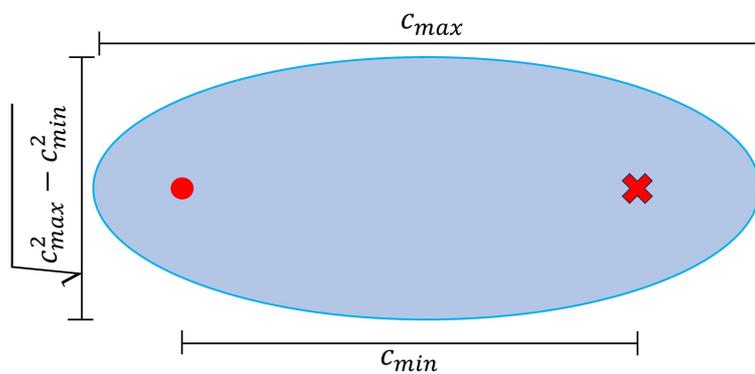
O Informed-RRT\* (e os algoritmos subsequentes que são baseados no Informed-RRT\*) é um planejador de caminhos amplamente consolidado e utilizado

Figura 13 – Metodologia Informed-RRT\*.



Fonte: Elaborado pelo Autor.

Figura 14 – Definição da Elipse no Informed-RRT\*.



Fonte: Retirado de [24].

em diversas aplicações recentes, incluindo aplicações com VANTS, conforme observado nos trabalhos [98, 99]. Devido às vantagens apresentadas por este algoritmo, a exploração autônoma neste trabalho será realizada por meio deste método.

### 3.3.2 Variante Direct-DRRT\*

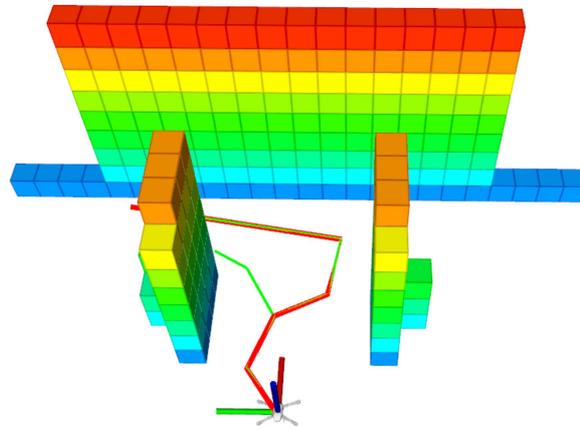
O Direct-DRRT\* é uma variante das metodologias baseadas no RRT, inicialmente proposta para robôs móveis [100] e posteriormente expandida para aplicações em ambientes 3D, como manipuladores robóticos [101]. Sua principal característica é a eficiência na entrega de trajetórias, destacando-se como alternativa à distância euclidiana, amplamente utilizada para definir custos entre pontos. Os resultados apresentados indicam tempos inferiores em relação às metodologias convencionais, tornando-a particularmente adequada para cenários onde a rapidez para a concepção de um caminho é vital. Neste trabalho, o Direct-DRRT\* será empregado como base para calcular o custo de deslocamento do drone durante a exploração autônoma, uma etapa essencial da metodologia proposta, detalhada na Seção 4.4. Essa escolha é motivada pela necessidade de avaliar múltiplas posições candidatas de forma eficiente, permitindo que o robô selecione rapidamente a melhor fronteira para explorar.

A metodologia apresenta um comportamento simples, realizando amostragens de forma convencional, similar ao RRT\*. No entanto, caso exista um caminho viável que conecte qualquer ponto da árvore ao objetivo, a expansão convencional da árvore é interrompida, e as amostragens passam a se concentrar na posição do objetivo (Característica *Direct*). A letra "D" no nome Direct-DRRT\* refere-se a uma heurística de amostragem fixa ao redor da posição do objetivo, que facilita a orientação da exploração de forma mais rápida e eficiente em direção ao ponto desejado [102].

Embora o método apresente tempos extremamente baixos para a obtenção de um caminho, as trajetórias geradas tendem a ser pouco otimizadas. No entanto, essa limitação não compromete a estimativa de custo entre dois pontos em um mapa, uma vez que a distância euclidiana — frequentemente utilizada em processos decisórios — geralmente desconsidera obstáculos presentes no ambiente, conforme ilustrado na Figura 15.

Na Figura 15, observa-se que o caminho gerado pelo Direct-DRRT\* realiza uma conversão abrupta em direção ao objetivo (linha vermelha), enquanto a distância euclidiana ignora a presença de obstáculos no ambiente. Apesar de o trajeto obtido por essa abordagem, baseada em RRT, não ser otimizado, ele não

Figura 15 – Exemplo de Caminhos Gerados pelo Direct-DRRT\*.



Fonte: Elaborado pelo Autor.

será empregado diretamente na navegação do VANT, mas sim como subsídio para a escolha da melhor posição objetivo. A árvore expandida durante a geração do caminho é representada pelos segmentos de linha exibidos na Figura 15 com a coloração verde.

## 4 METODOLOGIA PROPOSTA

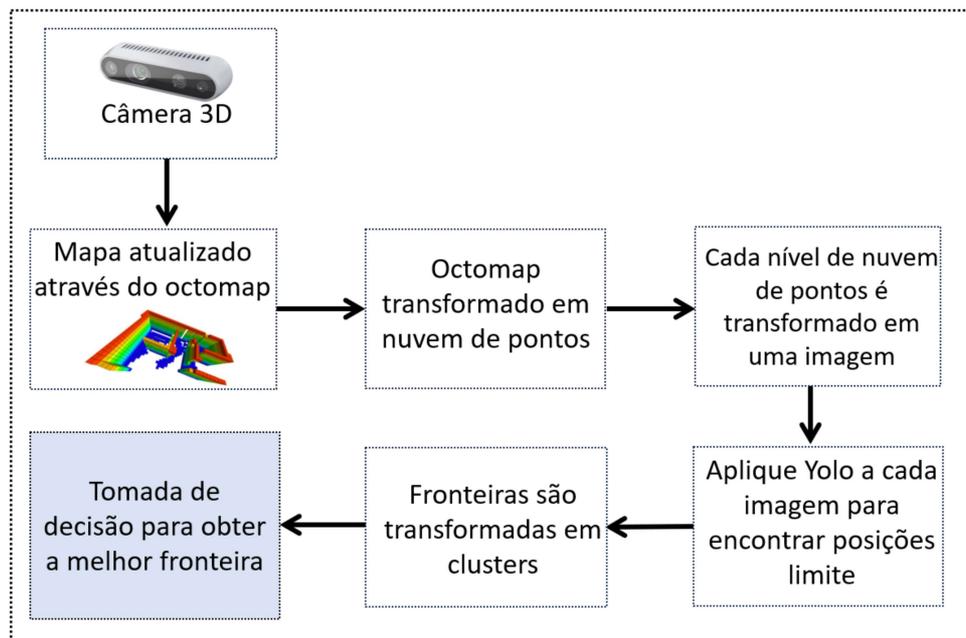
A metodologia desenvolvida nesta pesquisa propõe uma abordagem eficaz para a navegação autônoma de VANTs equipados com câmera 3D e sensores convencionais, com o objetivo de maximizar a eficiência da missão. O método busca enfrentar de forma integrada os principais desafios relacionados ao mapeamento, detecção de fronteiras, tomada de decisão e planejamento de rotas, permitindo que o robô explore o ambiente de maneira eficiente e adaptativa.

A proposta consiste, inicialmente, na criação e atualização contínua de um mapa tridimensional baseado na estrutura OctoMap, utilizando as informações de obstáculos captadas pela câmera 3D embarcada. Esse mapa 3D é posteriormente convertido em um conjunto de imagens, as quais são processadas por uma rede neural treinada para identificar fronteiras do ambiente explorado. As fronteiras detectadas são então agrupadas por meio de um algoritmo de clusterização, que consolida os segmentos em centroides com base na proximidade espacial. Na sequência, um algoritmo de tomada de decisão avalia cada centroide utilizando uma função de custo, também desenvolvida neste trabalho, para selecionar o objetivo mais apropriado. Por fim, um planejador de caminho traça uma rota composta por uma série de pontos que conectam a posição atual do robô ao centroide escolhido. Os detalhes de cada componente serão apresentados nas seções subsequentes, e a sequência metodológica está ilustrada na Figura 16.

### 4.1 EXTRAÇÃO DE DADOS DO AMBIENTE

Todas as missões são iniciadas com um mapa completamente vazio, no qual todos os pontos do ambiente são inicialmente classificados como desconhecidos. Os primeiros dados coletados são utilizados para atualizar esse mapa, marcando determinadas regiões como ocupadas (obstáculos) e outras como livres, de acordo com as observações obtidas no ambiente real. A coleta de dados de profundidade é realizada por meio da câmera Intel RealSense D435, fabricada pela Intel [103], que é capaz de capturar até 90 quadros por segundo. Esse dispositivo emprega um sensor estereoscópico para estimativa de profundidade e conta com um amplo campo de visão (*Field of View* - FoV) de  $87^\circ \times 58^\circ$ , o que o torna apropriado para

Figura 16 – Visão Geral da Metodologia.



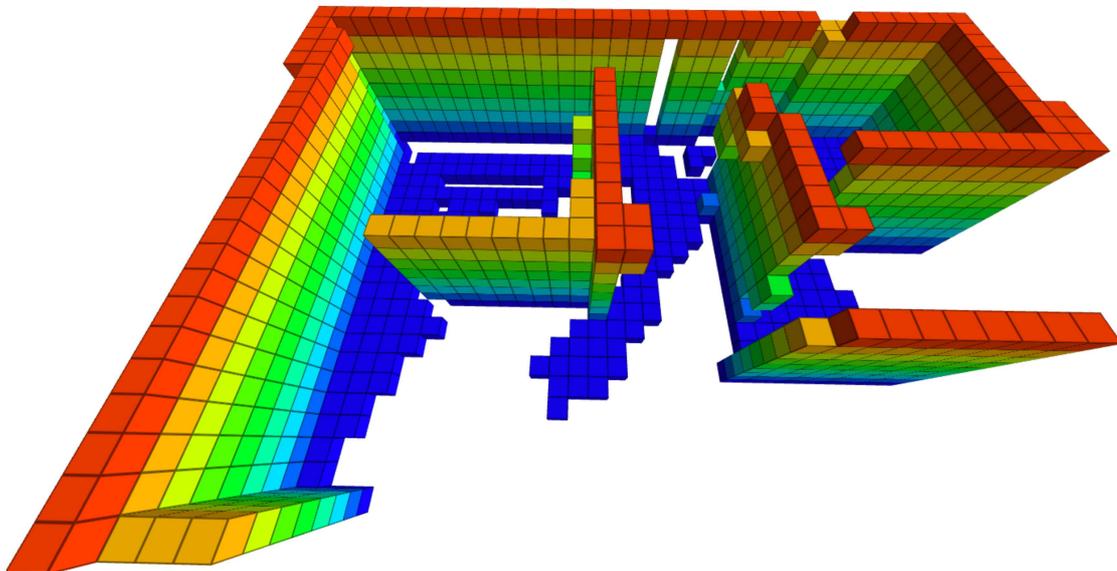
Fonte: Elaborado pelo Autor.

diversas aplicações com drones, tanto em ambientes internos [104, 105] quanto externos [106, 107].

O processo de mapeamento adotado baseia-se na metodologia OctoMap, conforme descrito na Seção 3.2, utilizando os dados tridimensionais obtidos pela câmera D435. A ocupação dos pontos no espaço é avaliada de forma probabilística, permitindo a distinção entre regiões livres e ocupadas com base em múltiplas observações. A Figura 17 ilustra um exemplo de OctoMap gerado durante uma missão de exploração. Nessa representação, o gradiente de cores reflete a elevação das regiões: os tons azulados correspondem às altitudes mais baixas, enquanto os tons avermelhados indicam regiões mais elevadas.

A Figura 17 apresenta um mapa com resolução de  $0.5\text{ m}$ , que será a mesma utilizada nas missões de exploração discutidas neste trabalho. Este valor foi definido de acordo com trabalhos encontrados na literatura que servirão como comparativos de desempenho em relação a nossa metodologia proposta. Considerando um mapa

Figura 17 – Exemplo de OctoMap Durante o Processo de Exploração.



Fonte: Elaborado pelo Autor.

onde a altura máxima de navegação de um drone seja de  $10\text{ m}$ , serão utilizados 20 níveis para representar o ambiente. Esse fatiamento do mapa 3D será responsável por criar imagens que categorizam as áreas como obstáculos, ambiente livre ou desconhecido. Estas imagens utilizam apenas três valores de pixel para representar essas categorias:

- Valor 0: *Grid* desconhecido;
- Valor 127: *Grid* sem obstáculo (Livre);
- Valor 255: *Grid* com um obstáculo (Ocupada).

Esta abordagem simplifica as questões interpretativas associadas às imagens capturadas por câmeras ópticas convencionais, facilitando a interpretação ao basear-se exclusivamente nos obstáculos visíveis durante o processo de exploração. A simplicidade das imagens terá um impacto positivo no aprendizado das técnicas de *machine learning* utilizadas neste trabalho. Essa modelagem representa um dos

aspectos inovadores deste estudo no que diz respeito à exploração e à identificação de fronteiras.

As dimensões do mapa 3D determinam o tamanho das imagens. Assim, uma imagem de alta resolução é necessária em mapas de grandes dimensões. No contexto do exemplo utilizado neste estudo, as dimensões da imagem são definidas como  $[60 \times 60]$ , com 20 imagens potenciais sendo extraídas do OctoMap. Isso indica que o mapa empregado possui dimensões de  $30m \times 30m$  e uma altura máxima de  $10m$ . Exemplos dessas imagens podem ser vistos na Figura 18, que representa o ambiente de teste. Para melhor visualização, um mapa de cores espectrais foi aplicado às imagens. As cores cinza, verde e preto representam *grids* desconhecidos, *grids* livres e *grids* com obstáculos, respectivamente.

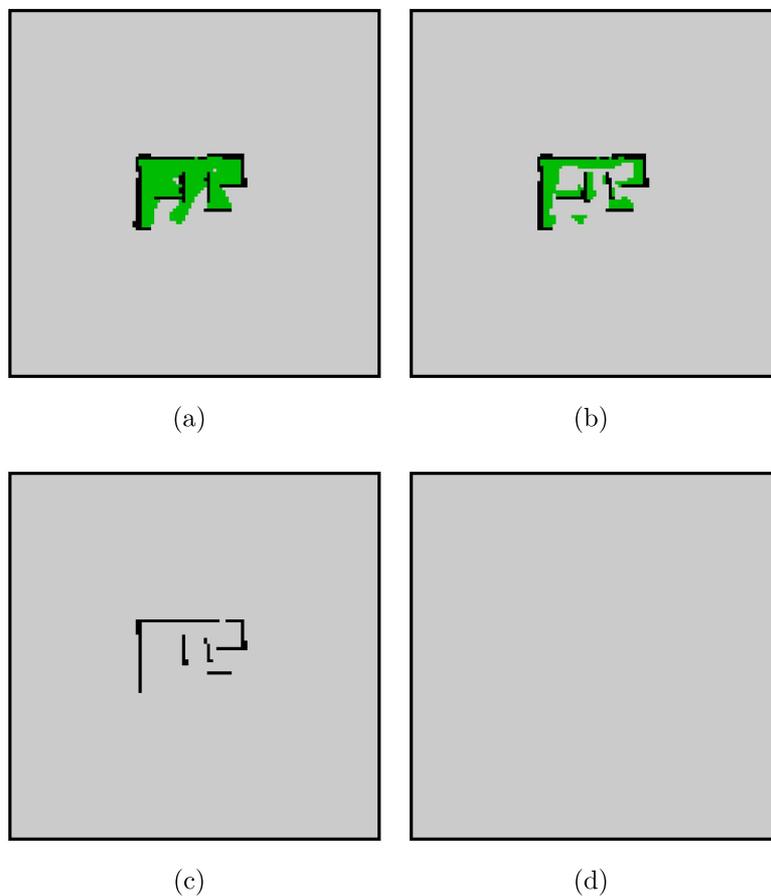
Ao analisar as imagens apresentadas na Figura 18, torna-se evidente uma das vantagens de nossa metodologia, especificamente na criação de imagens simplificadas que exibem um contraste notável entre áreas livres e desconhecidas, facilitando a definição de fronteiras.

Para gerar essas imagens, o drone manteve uma altitude de movimentação de 2 metros. Devido à natureza cônica das observações da câmera 3D, observa-se que em alturas mais elevadas, há um aumento na detecção de células desconhecidas. A partir do nível 9, a câmera deixou de detectar obstáculos ou regiões livres, resultando em que todas as células da grade são mantidas como desconhecidas (situação inicial). Isso indica claramente que o robô deve elevar-se e explorar regiões ainda inexploradas para expandir o mapa.

#### 4.2 SÍNTESE DO MODELO *DEEP LEARNING* PARA DETECÇÃO DE FRONTEIRAS

Conforme delineado na seção introdutória deste trabalho, as imagens em análise, ilustradas na Figura 18, mostram interações entre regiões cinzas e áreas verdes, representando, respectivamente, zonas desconhecidas e livres, cujo contato define o que é fronteiras. O próximo passo envolve a detecção das fronteiras do ambiente para orientar a navegação do robô. As fronteiras podem variar em forma e tamanho, sendo possível a coexistência de múltiplas fronteiras dentro de uma única

Figura 18 – Exemplos de Imagens Geradas: (a) Nível 1 ( $z = 0.5m$ ) (b) Nível 7 ( $z = 3.5m$ ) (c) Nível 8 ( $z = 4.0m$ ) (d) Nível 9 ( $z = 4.5m$ ).



Fonte: Elaborado pelo Autor.

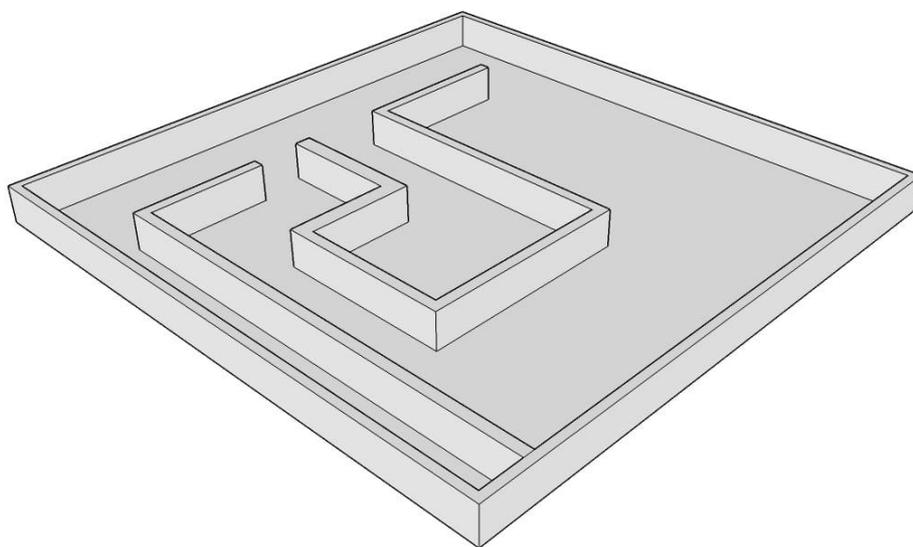
imagem. Além disso, é essencial localizar com precisão a posição das fronteiras para guiar o movimento do robô até elas. Essa detecção deve ocorrer em tempo real, a partir do OctoMap construído.

A problemática se encaixa adequadamente nas capacidades das metodologias baseadas no algoritmo YOLO para detecção rápida e eficaz. Portanto, o objetivo central torna-se a detecção das numerosas fronteiras presentes nas imagens extraídas na seção 4.1. Para este fim, foi empregada a versão 8 da YOLO, desenvolvida pela Ultralytics [70].

Para treinar a rede, foi estabelecido um banco de dados utilizando um

ambiente simulado, onde todas as movimentações do drone foram realizadas manualmente. Neste ambiente, foi realizada apenas a exploração manual com a intenção de obter imagens para que posteriormente seja anotado o banco de dados com as fronteiras existentes. O ambiente pode ser observado na Figura 19. Optou-se por empregar a versão *nano* da arquitetura YOLOv8, que possui a menor dimensionalidade. Conforme relatado pelos autores, esta escolha resultou em tempos de treinamento e predição de imagem significativamente reduzidos em comparação com outras arquiteturas.

Figura 19 – Mapa Utilizado para Criar o *Dataset*.



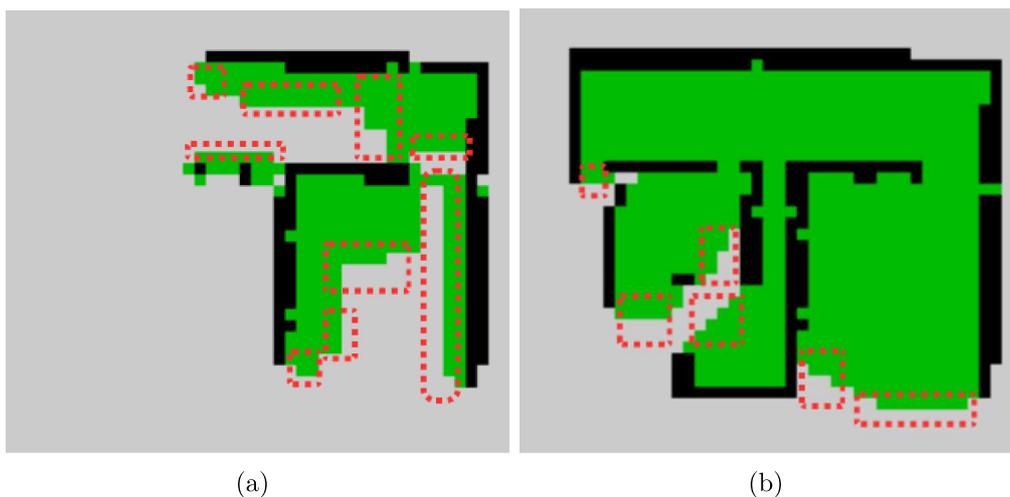
Fonte: Elaborado pelo Autor.

Para assegurar uniformidade, todas as imagens foram redimensionadas para uma resolução de  $605 \times 605$  pixels utilizando a técnica de interpolação de área fornecida pela biblioteca OpenCV [108]. Este aumento de resolução é crucial para auxiliar na detecção de objetos menores, um desafio que tem sido identificado em versões anteriores da YOLO.

A rede foi configurada para focar exclusivamente na detecção de fronteiras, constituindo uma única classe de detecção. O banco de dados resultante foi

composto por 200 imagens, todas anotadas no formato adequado para utilização com YOLO. A Figura 20 apresenta exemplos dessas anotações, destacando que as formas e tamanhos das fronteiras podem variar consideravelmente, dependendo da forma como o ambiente foi explorado.

Figura 20 – Exemplos de Fronteiras Para Anotação em Conjunto de Dados: (a) 9 Fronteiras (b) 6 Fronteiras.



Fonte: Elaborado pelo Autor.

As imagens exibidas na Figura 20 foram extraídas do mesmo nível do mapa usado para criar o banco de dados que, posteriormente, foi empregado para treinar a rede. A Figura 20(a) captura um momento da exploração ocorrido antes do registrado na Figura 20(b). As anotações, realizadas manualmente, mostram diferentes formatos de fronteiras, como mencionado anteriormente. As fronteiras encontradas são limitadas à quadriláteros.

As posições cartesianas  $x$  e  $y$  das fronteiras são determinadas pelos centros dos quadriláteros, enquanto a posição de altura é dada pelo nível correspondente da imagem. Assim, após o processamento da imagem pela estrutura de *deep learning*, o resultado é um conjunto de coordenadas  $x$ ,  $y$  e  $z$  que definem as fronteiras do ambiente. Esta tarefa é realizada em tempo real no cenário simulado. Entretanto, de acordo com os resultados apresentados em [70], é possível alcançar até 155 análises por segundo, o que resultaria em uma altura máxima de 75.2 m para a

resolução utilizada. Porém, caso seja necessário atingir alturas maiores, alterar a resolução do mapa pode ser uma solução mais indicada.

#### 4.2.1 Análise do Treinamento da Rede

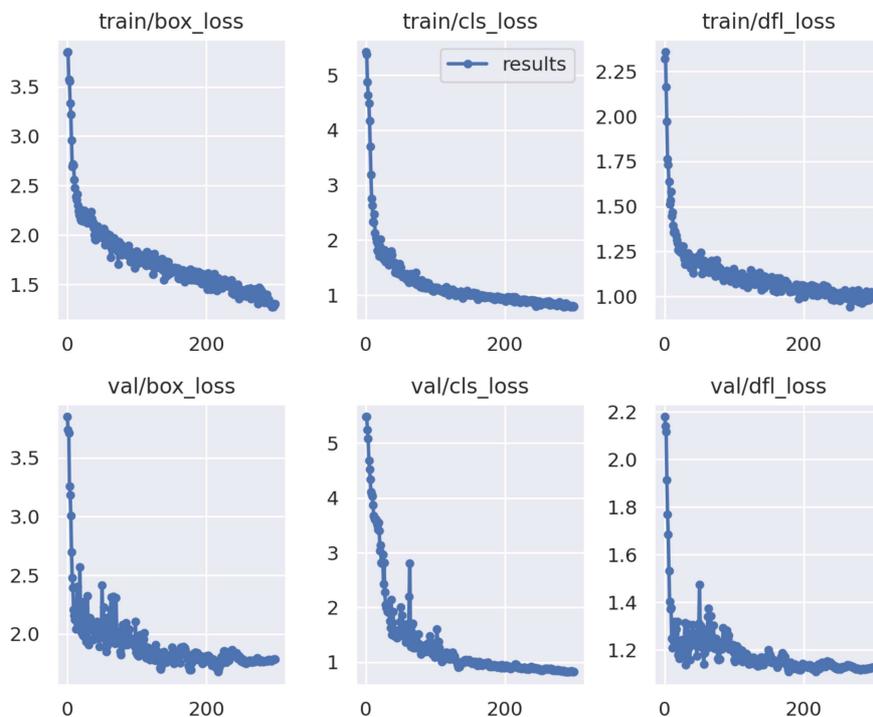
O modelo YOLOv8 foi treinado utilizando a arquitetura mais leve disponível (*YOLOv8n*) em sua versão pré-treinada, durante 300 épocas. Para garantir a generalização do modelo, empregaram-se as técnicas nativas de *data augmentation* da biblioteca Ultralytics, incluindo *flip* horizontal e vertical, transformações geométricas como rotação, translação e variação de escala, além do método *Mosaic* para combinação contextual de imagens. Optou-se por desativar o cisalhamento e as modificações no espaço de cores HSV (matiz, saturação e brilho), que não demonstraram contribuição significativa para a generalização do modelo em nosso domínio específico. O conjunto de dados foi dividido estrategicamente em 130 imagens para treinamento, 35 para validação e 35 para teste, assegurando uma avaliação abrangente do desempenho em cenários não vistos durante o treinamento.

Para o processo de treinamento, utilizou-se o algoritmo Gradiente Descendente Estocástico (*Stochastic Gradient Descent - SGD*) como método de otimização, executado integralmente em GPU durante as 300 épocas. O coeficiente de momento foi configurado como 0.937, desempenhando um papel crucial na aceleração da convergência ao direcionar o gradiente de forma mais eficiente durante o treinamento. A taxa de aprendizagem inicial foi estabelecida em 0.01, com um esquema dinâmico de decaimento que permitiu reduzir esse valor em até 10% do valor inicial ao longo do treinamento. O processamento foi realizado em *batches* de tamanho 32, balanceando eficiência computacional e estabilidade no treinamento. Como medida de regularização, adotou-se um coeficiente de decaimento de peso (L2) de 0.0005, efetivo na prevenção de *overfitting* - fenômeno caracterizado por alta performance no conjunto de treinamento que não se generaliza adequadamente para novos dados.

A convergência do processo de treinamento é quantificada pela redução progressiva das funções de perda, que apresentam comportamento assintótico aproximando-se de zero com o avanço das épocas. A Figura 21 apresenta a trajetória dessas métricas ao longo do treinamento, capturada durante a execução

da metodologia proposta cujo eixo vertical corresponde às perdas e o eixo horizontal às épocas.

Figura 21 – Avaliação Gráfica das Perdas Durante o Processo de Treinamento.



Fonte: Elaborado pelo Autor.

A análise da Figura 21 permite identificar três componentes fundamentais da função de perda: *box\_loss*, *cls\_loss* e *dfl\_loss*. A métrica *Box Loss* quantifica o erro de localização, medindo a discrepância entre as coordenadas preditas das caixas delimitadoras e os valores de *ground truth*. O *Class Loss* representa o erro de classificação, mesmo em cenários com classe única, onde o algoritmo YOLO incorpora automaticamente amostras de *background* como classe negativa para robustecer o aprendizado.

A *Distributed Focal Loss* (DFL) constitui um mecanismo sofisticado para tratamento de desequilíbrios de classes, particularmente relevante quando da inclusão de amostras de *background*. Esta função especializada opera em dois níveis: otimiza o posicionamento das caixas delimitadoras através de gradientes direcionais que

minimizam a distância para os valores reais, e atenua o impacto desproporcional de classes majoritárias no processo de aprendizado. A DFL demonstra especial eficácia na correção de falsos positivos, garantindo que as predições se alinhem precisamente com as anotações de referência.

Além da análise das funções de perda, a avaliação quantitativa do modelo requer o exame das métricas de precisão e *recall* obtidas durante o treinamento. No contexto de detecção de objetos baseada em *deep learning*, a precisão caracteriza a capacidade do modelo em realizar classificações corretas das regiões de fronteira, sendo formalmente definida pela Equação 4.1:

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (4.1)$$

onde *TP* (*True Positives*) representa o número de instâncias corretamente classificadas como pertencentes à classe de interesse, e *FP* (*False Positives*) denota os casos em que regiões sem o objeto de interesse foram erroneamente classificadas como positivas.

A sensibilidade do modelo, denominada *recall*, quantifica sua capacidade de detectar todas as instâncias relevantes da classe alvo, considerando especialmente os falsos negativos. Esta métrica é formalizada pela Equação 4.2:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4.2)$$

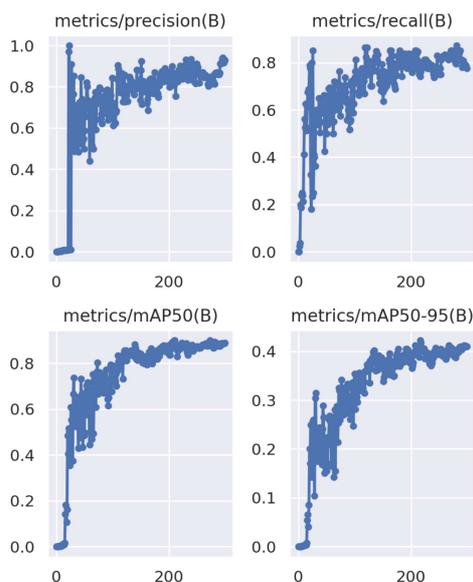
sendo *FN* (*False Negatives*) o número de objetos de interesse que não foram detectados pelo modelo. Essas duas métricas complementares fornecem uma avaliação abrangente do desempenho do modelo, considerando tanto os erros de falsos positivos quanto os erros de falsos negativos.

A avaliação da qualidade das caixas delimitadoras é realizada por meio da métrica *mean Average Precision* (mAP), considerada padrão-ouro para avaliação de modelos de detecção de objetos. Esta métrica quantifica a precisão média do modelo considerando todas as classes detectáveis, mediante a aplicação de um limiar de IoU para análise da sobreposição espacial entre as caixas preditas e as anotações de referência.

A métrica mAP50, especificamente, avalia o desempenho do modelo utilizando um limiar mínimo de IoU de 50%, representando o requisito básico de alinhamento espacial. Para uma análise mais rigorosa, emprega-se a métrica mAP50-95, que calcula a média das precisões médias em intervalos de 5% do limiar de IoU, variando de 50% a 95%. Esta abordagem multidimensional permite caracterizar de forma abrangente a capacidade do modelo em gerar previsões precisas tanto para detecções gerais quanto para localizações refinadas de objetos.

A Figura 22 apresenta os resultados quantitativos desta análise, demonstrando a eficácia do modelo na tarefa de detecção de fronteiras em diferentes níveis de exigência de precisão espacial. A interpretação desses valores é fundamental para avaliar o equilíbrio entre sensibilidade e precisão do detector em diferentes cenários operacionais. Enquanto que o eixo vertical corresponde aos índices com valor máximo 1, o eixo horizontal corresponde às épocas do treinamento.

Figura 22 – Métricas de Desempenho Obtidas no Treinamento.



Fonte: Elaborado pelo Autor.

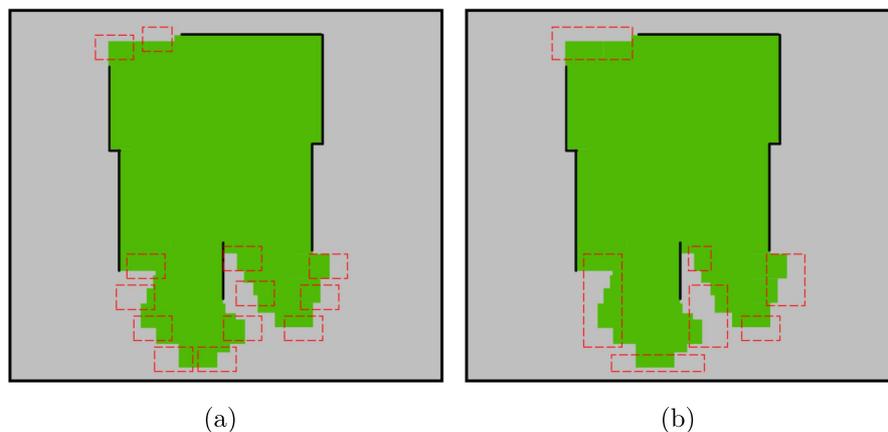
O modelo alcançou valores de mAP50 e mAP50-95 de 89% e 41%, respectivamente. Para comparação, quando a YOLOv8n foi treinada para detectar os

vários objetos do banco de dados COCO [109], ele registrou um mAP50-95 de 37%. Apesar dos desafios impostos pelos formatos e tamanhos variados que podem ser caracterizados como fronteiras, nosso modelo demonstrou desempenho superior em comparação com aos resultados em detecção de objetos apresentados em [70]. Esses resultados não apenas justificam o uso de *deep learning* para a detecção de fronteiras, mas também representarão um avanço significativo no campo da exploração autônoma.

### 4.3 CLUSTERIZAÇÃO DE FRONTEIRAS

Após a predição de fronteiras a partir das inúmeras imagens previamente geradas no fatiamento do mapa 3D, é necessário avaliar os resultados obtidos para transmitir ao robô coordenadas que sejam factíveis com a realidade. Um dos primeiros pontos observados é a infinidade de fronteiras que podem ser identificadas em uma única inferência, permitindo que os agrupamentos (clusters) sejam consolidados de diversas maneiras. Na Figura 23, é possível observar como essas fronteiras surgem após uma predição.

Figura 23 – (a) Arranjo de Fronteiras 1 (b) Arranjo de Fronteiras 2.



Fonte: Elaborado pelo Autor.

A partir da Figura 23, é possível observar que, para uma mesma imagem, pode-se definir diversos arranjos de fronteiras. É evidente que considerar essas fronteiras de forma isolada pode comprometer a exploração eficiente do ambiente.

Portanto, é necessário realizar um processo de clusterização para identificar pontos que representem múltiplas fronteiras. Além disso, algumas fronteiras podem ser vistas como redundantes, uma vez que níveis acima e abaixo de uma determinada imagem frequentemente formam uma única fronteira contínua no ambiente. A Figura 24 ilustra os resultados obtidos em diferentes níveis na conversão do OctoMap para imagens.

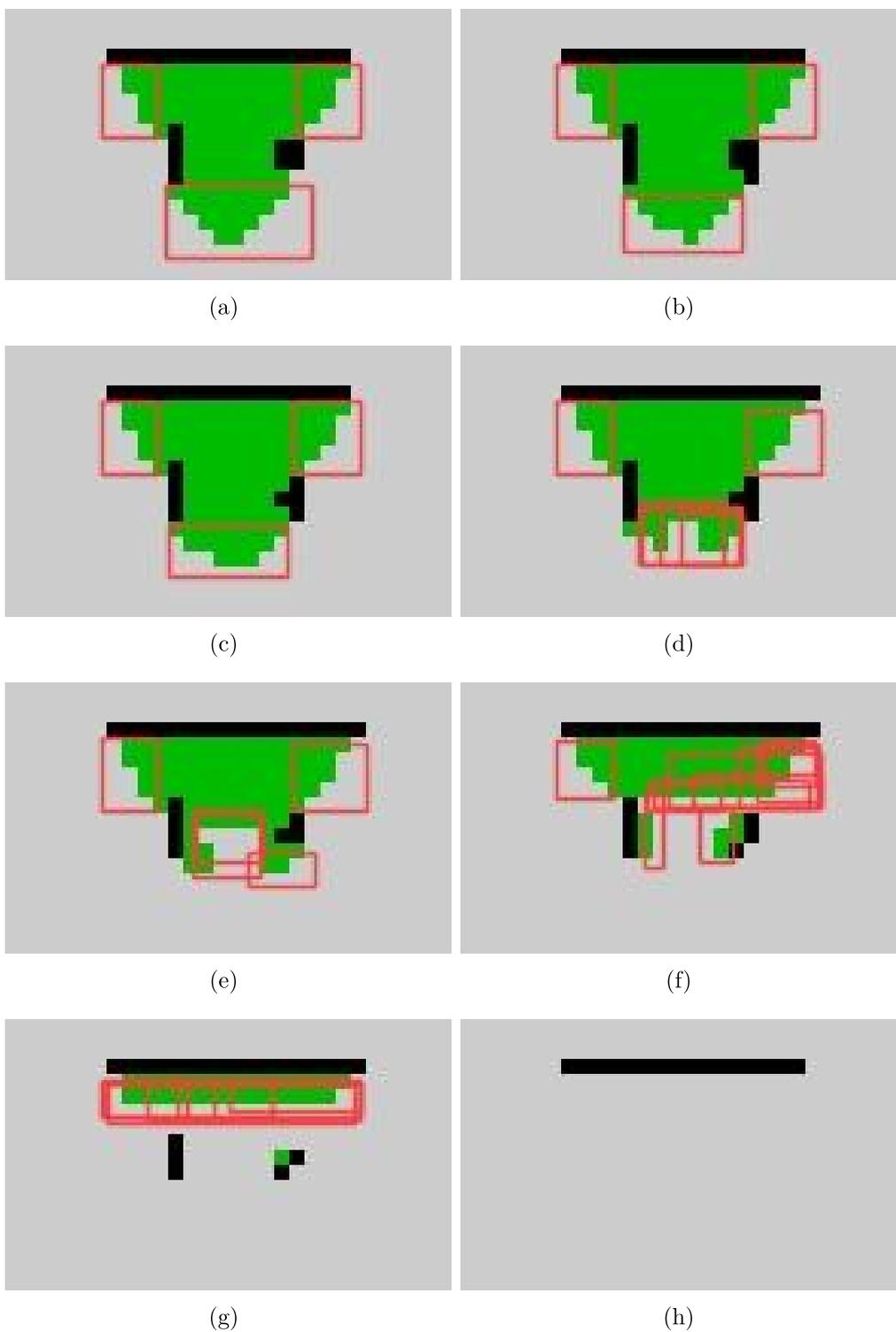
A necessidade de clusterização pode ser observada nas Figuras 24(a), (b) e (c), onde as três fronteiras detectadas correspondem ao mesmo conjunto de fronteiras, diferenciando-se apenas pela altura. A Figura 24(d) também mostra várias regiões de fronteiras muito próximas entre si (sobreposição de áreas), que, após análise, confirmam pertencer ao mesmo conjunto. Esta sobreposição e redundância de fronteiras têm impacto direto na tomada de decisão do robô em relação a qual direção seguir, pois fronteiras duplicadas ou regiões redundantes podem influenciar significativamente essa escolha.

Para a técnica de agrupamento, foi utilizado o algoritmo K-Means [110] juntamente com o algoritmo de Lloyd, que adota uma abordagem empírica probabilística para agrupar as fronteiras em regiões similares [111]. O algoritmo K-Means é um método de *machine learning* não supervisionado, utilizado em situações onde os dados não possuem rótulos previamente definidos. Assim, o algoritmo opera de modo a separar os dados de forma inteligente. Ele inicia distribuindo  $k$  (quantidade de clusters pré-definido) centróides pelo mapa de maneira aleatória ou conforme uma estratégia pré-definida. Estes centróides servem como referências para a caracterização dos clusters.

A estratégia de inicialização dos centróides adotada neste trabalho é o *K-Means++ greedy* [112], que detecta de modo empírico a distribuição de probabilidade das amostras. Esta abordagem visa posicionar os centróides inicialmente de maneira que eles estejam mais próximos das amostras que eventualmente descreverão os clusters.

Uma vez definido o primeiro conjunto de centróides, cada amostra é associada a um centróide específico de acordo com uma métrica pré-definida. Neste trabalho, utilizou-se a distância euclidiana entre cada amostra e cada centróide para determinar a associação mais apropriada.

Figura 24 – Fronteiras Encontradas no: (a) Nível 1. (b) Nível 2. (c) Nível 3. (d) Nível 4. (e) Nível 5. (f) Nível 6. (g) Nível 7. (h) Nível 8.



Fonte: Elaborado pelo Autor.

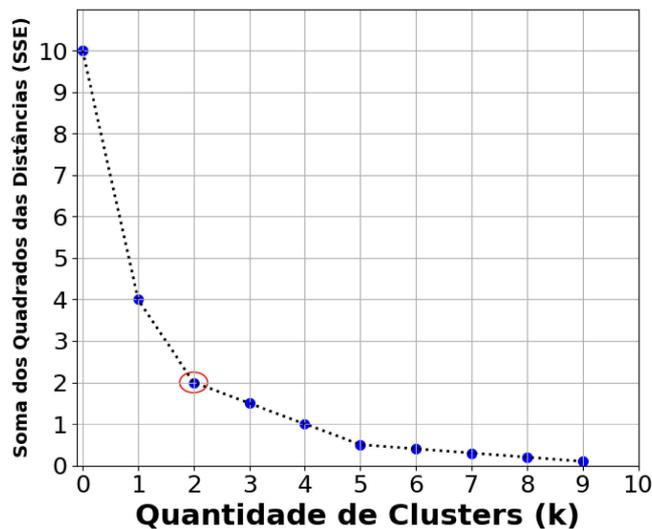
A iteratividade do K-Means consiste na atualização dos centróides, que é realizada calculando a média aritmética das posições das amostras associadas a cada centróide. Em cada iteração, os centróides são atualizados e uma nova rodada de atribuição de amostras ocorre. O algoritmo termina quando nenhuma amostra muda de cluster ou quando o movimento dos centróides entre iterações não é significativo.

O algoritmo K-Means pode tornar-se ineficiente caso a quantidade de dados seja grande ou quando é necessário aplicar o algoritmo para diferentes valores de  $k$ . Por isso, adotamos estratégias que mantêm os valores dos centróides previamente computados ao longo da metodologia de exploração. Como a detecção de fronteiras ocorre em tempo real, espera-se que não ocorram grandes alterações nas posições dos centróides, o que, por sua vez, reduz o tempo necessário para a convergência do K-Means.

Embora a aplicação do algoritmo K-Means requeira que o número de clusters ( $k$ ) seja conhecido antecipadamente, a determinação precisa de  $k$  é desafiadora em nosso contexto, pois não se sabe quantas fronteiras serão encontradas ao explorar um ambiente. Portanto, foi realizada uma análise para determinar a quantidade de clusters necessária para segmentar adequadamente as inúmeras fronteiras identificadas. Para isso, o K-Means foi aplicado a vários valores de  $k$ , e uma métrica específica (i.e., SSE) foi utilizada para avaliar se a quantidade de clusters escolhida é ótima. Esta métrica é a soma dos quadrados das distâncias de cada amostra até seus centróides. Um exemplo de uma curva que representa o comportamento desta métrica, que auxiliará na seleção do  $k$  ideal, pode ser visualizada na Figura 25.

A análise da Figura 25 revela que, após atingir um certo número de clusters representado por  $k$  (marcado em vermelho), não há ganhos significativos no valor da soma dos quadrados das distâncias (SSE). Portanto, aumentar a quantidade de divisões além desse ponto pode comprometer a eficiência da exploração autônoma. O ponto em que não há mais ganho é conhecido como cotovelo (*elbow*) ou joelho (*knee*).

Embora a análise visual da soma dos quadrados das distâncias (SSE) para vários valores de  $k$  seja simples, é essencial automatizar essa tarefa para integrá-la

Figura 25 – Ponto de Inflecção para Avaliação de  $k$ .

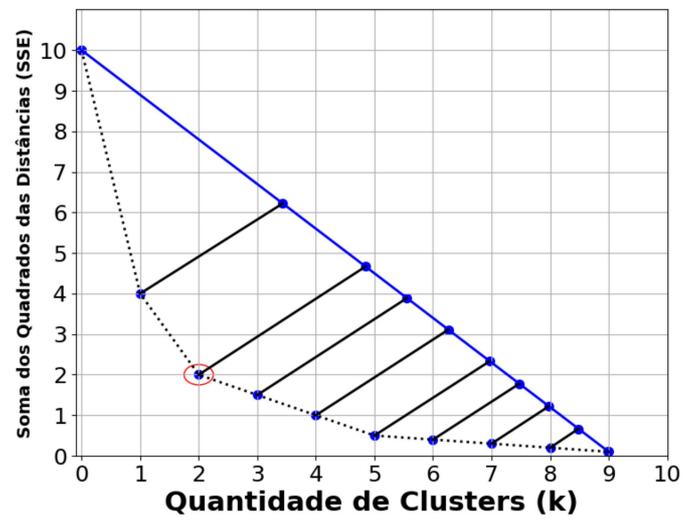
Fonte: Elaborado pelo Autor.

à nossa metodologia de exploração autônoma. Métodos como o índice de Silhueta, que avalia a qualidade dos clusters formados [113], e o algoritmo Kneedle [114] foram desenvolvidos como soluções inteligentes e eficientes para determinar o valor ótimo de  $k$ . Este valor é crucial, pois indica o número de clusters necessário para segmentar adequadamente as fronteiras detectadas, já que o objetivo é sempre buscar a maior quantidade de fronteiras possíveis onde ocorra um real ganho de informação.

Inicialmente, o algoritmo normaliza os dados de SSE e os valores de  $k$  para facilitar a comparação. Posteriormente, traça-se um gráfico similar ao apresentado na Figura 25, onde uma linha é desenhada conectando o primeiro e o último valor de  $k$ . O valor ótimo de  $k$  é identificado como aquele que possui a maior distância vertical em relação a essa linha. A ilustração deste algoritmo é representado na Figura 26.

A Figura 27 ilustra as fronteiras detectadas e seu agrupamento subsequente, representado por esferas pretas. É possível observar que, após o processo de

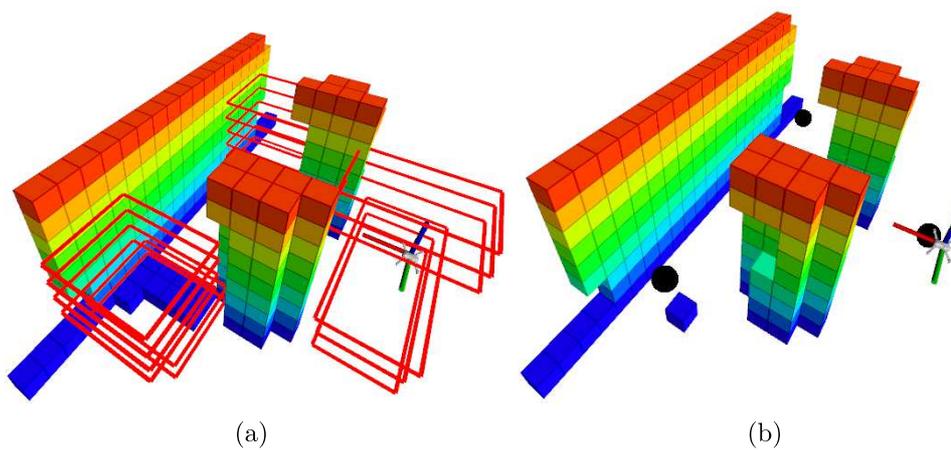
Figura 26 – Metodologia Kneedle para Detecção do Ótimo  $k$ .



Fonte: Elaborado pelo Autor.

clusterização, as fronteiras foram simplificadas em apenas três regiões distintas.

Figura 27 – Clusterização: (a) As Fronteiras São Representadas Por Retângulos Vermelhos. (b) Clusters Oriundos do K-Means são Representados por Esferas Pretas.



Fonte: Elaborado pelo Autor.

Um aspecto importante inerente à natureza da arquitetura YOLO é sua capacidade de reconhecer objetos contidos dentro de outros objetos. Esse comportamento é regulado pelo parâmetro IoU na função de predição, o qual define a porcentagem máxima de sobreposição permitida entre duas ou mais detecções. Valores mais baixos de IoU resultam em uma menor quantidade de fronteiras identificadas, o que contribui para a redução de detecções duplicadas. A ocorrência de duplicidades pode ser claramente observada nas Figuras 24(f) e (g), onde múltiplas detecções redundantes são geradas para a mesma região.

Esse problema é atenuado com a aplicação do filtro de *Non-Maximum Suppression* (NMS), que elimina predições redundantes ao manter apenas aquela com maior confiança. Ressalta-se que, caso regiões extensas apresentem sobreposição significativa, isso pode levar a uma tomada de decisão enviesada quanto ao cluster mais apropriado para o deslocamento do drone, comprometendo a eficiência da navegação.

#### 4.4 TOMADA DE DECISÃO NA SELEÇÃO DA MELHOR FRONTEIRA

Conforme ilustrado na Figura 27(b), o conjunto de fronteiras detectadas é particionado em clusters espaciais que representam regiões candidatas para navegação, com o objetivo principal de expansão sistemática do mapa explorado. Este processo de clusterização culmina em uma etapa crítica de seleção de alvos (problema de "tomada de decisão"), onde se determina qual cluster deve ser priorizado como objetivo de navegação para maximizar a eficiência da exploração.

A política de seleção de clusters ótimos fundamenta-se em critérios de decisão baseados no estado atual do ambiente mapeado, seguindo os princípios estabelecidos por [115] e [28]. Formalmente, uma maneira de selecionar o cluster ótimo ( $c_{best}$ ) pode ser observada pela equação de política de decisão apresentada na Equação 4.3, que integra múltiplos fatores espaciais e de informação do mapa para otimizar o processo exploratório.

$$c_{best} = \arg \max_{c_i \in \mathcal{C}} (A(c_i) * e^{(-\lambda L(P_{c_i}, P_{robot}))}) \quad (4.3)$$

A Equação 4.3 é utilizada para selecionar o valor máximo calculado entre

todos os clusters  $i$  pertencentes ao conjunto de clusters  $C$ . A função  $A(c_i)$  estima o possível ganho na expansão do mapa caso o cluster  $i$  seja escolhido. Esta função baseia-se na quantidade de *grids* desconhecidos ao redor do cluster. Por outro lado, a função  $L(P_{c_i}, P_{robot})$  calcula a distância euclidiana entre a posição atual do robô ( $P_{robot}$ ) e a posição do cluster ( $P_{c_i}$ ). Desta forma, um cluster muito distante do robô pode não ser a melhor escolha. O equilíbrio entre as funções  $A$  e  $L$  é regulado por uma constante  $\lambda$ . O cluster que alcançar a pontuação mais alta dentro do conjunto  $C$  será o escolhido para exploração.

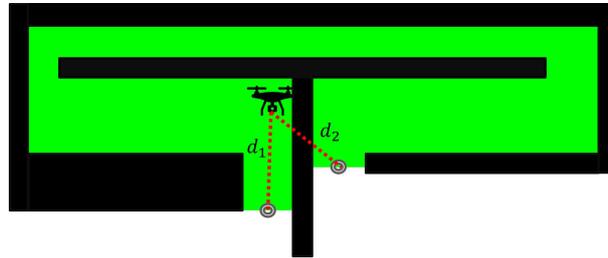
Este trabalho propõe uma nova política de tomada de decisão que supera limitações inerentes à abordagem apresentada na Equação 4.3. Esta formulação inicia-se com uma reformulação fundamental do custo de deslocamento entre a posição atual do robô e os candidatos a objetivo, substituindo a métrica euclidiana por um planejador de trajetórias que considera explicitamente a configuração espacial dos obstáculos, as restrições cinemáticas do robô e a viabilidade dinâmica do movimento. Esta abordagem gera estimativas de custo mais realistas, pois refletem trajetórias executáveis pelo sistema robótico.

As limitações da métrica euclidiana tornam-se particularmente evidentes em ambientes estruturados com geometria complexa, conforme ilustrado na Figura 28. Em cenários onde o caminho direto é obstruído, a distância euclidiana subestima significativamente o custo real de navegação, podendo levar à seleção de objetivos aparentemente próximos mas de difícil acesso. Nossa abordagem mitiga este problema através da integração de um modelo de custo baseado em planejamento de caminhos, garantindo que as decisões de exploração considerem a acessibilidade real de cada cluster.

A Equação 4.3 estabelece que a seleção das fronteiras ótimas considera a distância euclidiana entre o robô e as fronteiras detectadas de forma inversamente proporcional, e proporcional à contagem de *grids* desconhecidos ao redor de cada cluster. Contudo, como ilustrado na Figura 28, esta abordagem pode levar a decisões subótimas em ambientes complexos.

O cenário apresenta duas fronteiras ( $d_1$  e  $d_2$ ) com áreas similares de *grids* desconhecidos, onde  $d_2$  possui menor distância euclidiana em relação à posição atual do robô. Neste caso, a política de decisão baseada exclusivamente na métrica

Figura 28 – Problemas Observados ao se Utilizar a Distância Euclidiana como Custo de Deslocamento.



Fonte: Elaborado pelo Autor.

euclidiana selecionaria  $d_2$  como alvo preferencial. Entretanto, uma análise mais rigorosa que considere o caminho navegável real - levando em conta a configuração dos obstáculos - revela que a fronteira  $d_1$  pode ser significativamente mais acessível.

Este fenômeno evidencia uma limitação fundamental da abordagem euclidiana: a incapacidade de incorporar a complexidade topológica do ambiente no processo decisório. Fronteiras aparentemente próximas podem exigir trajetórias substancialmente mais longas e complexas devido à presença de obstáculos, enquanto alternativas com áreas de exploração equivalentes mas mais distantes na métrica euclidiana podem oferecer caminhos mais diretos e eficientes. Nossa proposta aborda esta limitação através da integração de um modelo de custo baseado em planejamento de trajetórias factíveis.

Para superar esta limitação, adotamos a metodologia Direct-DRRT\* para cálculo do custo de deslocamento até as fronteiras, conforme detalhado na Seção 3.3.2. A escolha deste planejador de trajetórias justifica-se por sua eficiência computacional, essencial para aplicações em tempo real onde múltiplas fronteiras candidatas devem ser avaliadas simultaneamente. Esta característica é particularmente relevante para sistemas robóticos autônomos, como drones, que exigem decisões rápidas para maximizar a eficiência exploratória. O custo calculado pelo método é formalizado pela função  $R(P_{c_i}, P_{robot})$ , onde  $P_{c_i}$  representa a posição candidata e  $P_{robot}$  a posição atual do robô, sendo ponderado pelo coeficiente  $\lambda_R$  para regular sua contribuição na função objetivo global.

Nossa abordagem incorpora ainda um módulo de inteligência artificial para análise avançada das fronteiras, substituindo a métrica convencional baseada em *grids* desconhecidos adjacentes ( $A$ ). As abordagens tradicionais avaliam o potencial exploratório mediante um raio fixo ao redor do cluster, o que frequentemente resulta em duas limitações críticas: desconsideração de áreas relevantes de fronteira fora do raio pré-definido, e avaliação inadequada de clusters com geometria complexa. Para resolver estas questões, introduzimos a função  $DL(c_i)$  (Equação 4.4), que quantifica o potencial exploratório de cada cluster  $c_i$  através de dois fatores principais: a distribuição espacial das fronteiras constituintes e suas características geométricas individuais. Esta medida proporciona uma avaliação mais precisa do ganho exploratório, superando as limitações inerentes aos métodos baseados exclusivamente em contagem discreta de *grids*.

$$DL(c_i) = \sum_{j=1}^{n_i} (l \times h)_j \quad (4.4)$$

Todas as fronteiras  $n$  contidas no cluster  $i$  são representadas por retângulos com largura  $l$  e altura  $h$ . A soma das áreas desses retângulos são agregadas para calcular o valor da informação proveniente da detecção de objetos realizada pela YOLO, a qual é uma componente essencial de nossa metodologia. A influência da distância entre a posição atual e a posição do cluster é ponderada por meio de uma curva exponencial negativa na Equação 4.3, de modo que quanto maior a distância, menor será a pontuação atribuída ao cluster. Por outro lado, para considerar a importância dos clusters de forma proporcional à área prevista pela rede neural, optamos por incorporar uma função exponencial positiva. Para garantir uma comparação justa entre os diferentes conjuntos de dados, normalizamos as informações de distância e área. Dessa maneira, a fórmula para a tomada proposta pode ser visualizada na Equação 4.5.

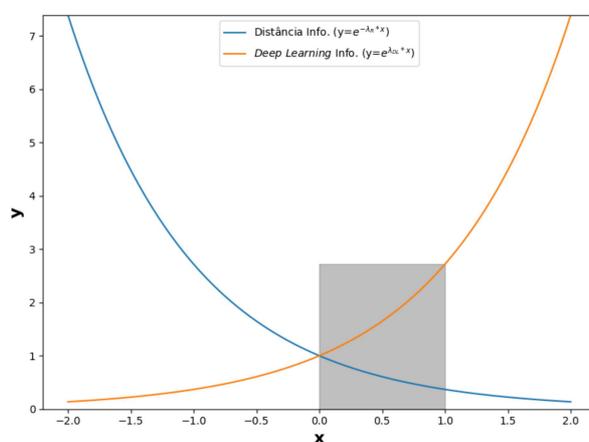
$$c_{best} = \arg \max_{c_i \in \mathcal{C}} (e^{(-\lambda_R R(P_{c_i}, P_{robot}) + \lambda_{DL} DL(c_i))}) \quad (4.5)$$

Observa-se que a função  $A(c_i)$  foi descontinuada e substituída pela informação  $DL(c_i)$ , que apresenta resultados significativamente mais precisos. A nova equação, apresentada em 4.5, introduz dois parâmetros ajustáveis:  $\lambda_R$  e  $\lambda_{DL}$ . Esses

parâmetros foram concebidos para controlar o impacto relativo de cada componente na determinação do custo final.

Entretanto, é essencial que esses valores sejam escolhidos com cuidado, pois uma seleção inadequada pode fazer com que uma das informações prevaleça excessivamente sobre a outra, distorcendo o cálculo dos custos. A Figura 29 ilustra os gráficos das exponenciais negativas e positivas, demonstrando a abordagem adotada na seleção das melhores fronteiras. Para este gráfico, os valores de  $\lambda_R$  e  $\lambda_{DL}$  foram definidos como 1 (mesmo valor), equilibrando de maneira uniforme a influência de ambos os parâmetros na equação.

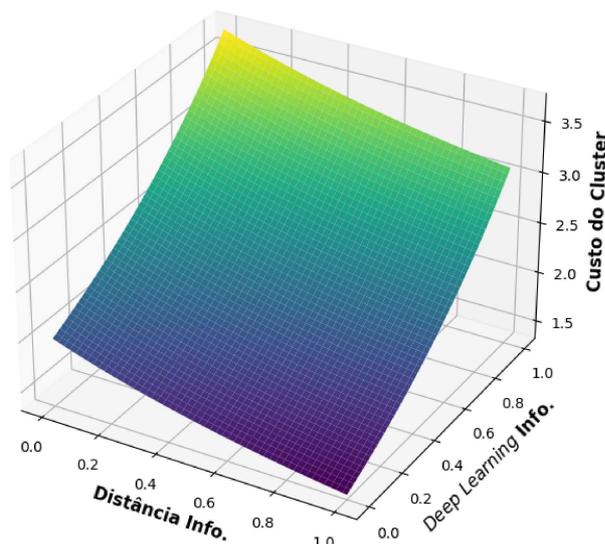
Figura 29 – Curvas Exponenciais Negativa e Positiva.



Fonte: Elaborado pelo Autor.

Analisando a Figura 29, observa-se que a região correspondente à exponencial negativa apresenta valores menores, enquanto a região da exponencial positiva exibe valores superiores. A área sombreada em cinza representa os valores normalizados de distância e área utilizados ao longo da exploração. Portanto, a menos que se deseje dar mais peso às informações provenientes do modelo de rede neural, o ganho associado à distância deve ser superior ao ganho  $\lambda_{DL}$ . Essa discrepância nos custos é mais claramente visualizada no gráfico 3D ilustrado na Figura 30, onde é demonstrada a situação em que  $\lambda_R = \lambda_{DL}$ .

A partir da Figura 30, observa-se que o custo associado à maior fronteira

Figura 30 – Custo dos Clusters Quando  $\lambda_R = \lambda_{DL}$ .

Fonte: Elaborado pelo Autor.

será superior ao custo correspondente à menor distância. Assim, para equalizar a importância das duas informações utilizadas na Equação 4.5, é necessário reduzir a influência do valor da informação oriunda da rede neural através da constante reguladora  $\lambda_{DL}$ . O valor ideal de  $\lambda_{DL}$  para equalizar as ponderações pode ser determinado pela Equação 4.6.

$$e^{(-\lambda_R*d_1)} + e^{(\lambda_{DL}*a_1)} = e^{(-\lambda_R*d_2)} + e^{(\lambda_{DL}*a_2)} \quad (4.6)$$

É importante notar que o objetivo é alcançar o mesmo valor de custo quando a distância/área são mínimas ( $d_1$  e  $a_1$  iguais a 0), assim como nos casos em que a distância/área são máximas ( $d_2$  e  $a_2$  iguais a 1). Para simplificar, vamos definir  $\lambda_R$  como 1 e o resultado desejado de  $\lambda_{DL}$  será proporcional a  $\lambda_R$ . Portanto, para resolver a equação  $e^{-0} + e^{0*\lambda_{DL}} = e^{-1} + e^{1*\lambda_{DL}}$ , fazemos:

$$e^{-0} + e^{0*\lambda_{DL}} = e^{-1} + e^{1*\lambda_{DL}}$$

$$1 + e^0 = e^{-1} + e^{\lambda_{DL}}$$

$$1 + 1 = \frac{1}{e} + e^{\lambda_{DL}}$$

$$2 = \frac{1}{e} + e^{\lambda_{DL}}$$

Agora, podemos resolver para  $\lambda_{DL}$ :

$$e^{\lambda_{DL}} = 2 - \frac{1}{e}$$

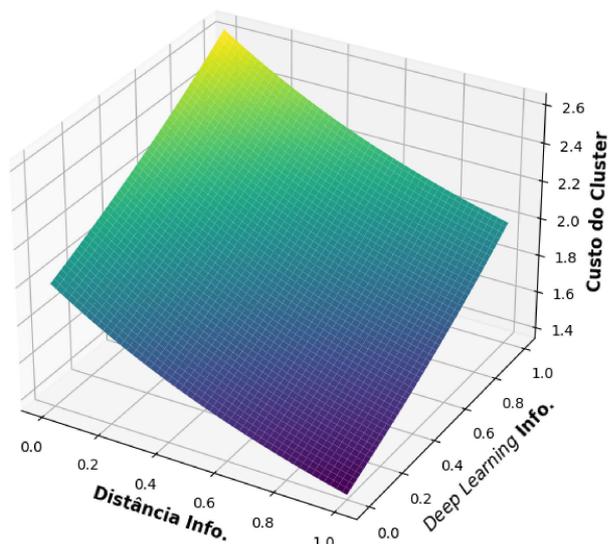
Tomando o logaritmo natural dos dois lados, obtemos:

$$\lambda_{DL} = \ln\left(2 - \frac{1}{e}\right)$$

O valor ótimo  $\lambda_{DL} = 0.48988$  foi determinado como uma das soluções exatas de balanceamento proposta neste trabalho. Esta constante representa o ponto de equilíbrio matemático que equaliza adequadamente os pesos das duas informações no processo de tomada de decisão para as pré-condições definidas anteriormente ( $\lambda_R = 1$ ). A solução analítica garante propriedades ótimas de estabilidade neste quesito para o sistema autônomo, conforme demonstrado nas missões experimentais realizadas. É importante destacar que, embora o valor  $\lambda_{DL} = 0.48988$  tenha sido derivado matematicamente como solução ótima para as condições padrão de balanceamento, a formulação admite ajustes quando requisitos específicos da missão assim o exigirem. A Figura 31 apresenta a superfície de desempenho correspondente a este valor, evidenciando as características de balanceamento e simetria do sistema de tomada de decisão quando operando com este parâmetro.

É importante ressaltar que a tomada de decisão não garante a escolha absolutamente ótima, uma vez que as decisões são baseadas nas informações disponíveis para o robô no momento da escolha. Portanto, é possível que, após a exploração de um cluster, se constate que a quantidade de mapa aberto seja menos eficiente do que outro cluster presente no mapa. Nesse sentido, a afirmação

Figura 31 – Custo dos Clusters Quando  $\lambda_R = \lambda_{DL} * 0.48988$ .



Fonte: Elaborado pelo Autor.

correta é que o robô toma a melhor decisão considerando o conjunto de informações disponíveis a ele no momento.

#### 4.5 ALGORITMO DE EXPLORAÇÃO AUTÔNOMA BASEADO EM FRONTEIRAS UTILIZANDO *DEEP LEARNING*

Neste capítulo, foi possível observar como cada um dos algoritmos utilizados será responsável por guiar o robô durante o processo de exploração autônoma. Esse processo ocorre de maneira iterativa, e sua representação pode ser observada por meio do algoritmo apresentado em 1.

O algoritmo é inicializado definindo um mapa baseado na estrutura do OctoMap com uma resolução  $r$  fixa para ser utilizada em todo o processo de exploração. Nesta fase, também são criados conjuntos de imagens, diretamente relacionados à resolução do OctoMap. A quantidade de níveis de imagens necessários corresponde à relação entre a altura máxima permitida para exploração e a resolução

---

**Algoritmo 1** Exploração Autônoma
 

---

- 1: Inicializa a estrutura de OctoMap  $O$  com resolução  $r$
  - 2: Inicializa um conjunto de  $n$  imagens
  - 3: **Inicia a Rotina Paralela**
  - 4: **Enquanto** Missão Autônoma Acontece **faça**
  - 5:     Extraí as leituras  $s$  da câmera
  - 6:     Atualiza o OctoMap  $O$  com as leituras  $s$
  - 7: **Fim do Enquanto**
  - 8: **Finaliza a Rotina Paralela**
  - 9: **Enquanto** Volume total (%) não for explorado **faça**
  - 10: **Operações Sequenciais:**
  - 11:     Conversão do OctoMap  $O$  em  $n$  imagens
  - 12:     Aplicação do modelo *deep learning* nas  $n$  imagens para obter as fronteiras  $f$
  - 13:     Pós-processamento das fronteiras  $f$
  - 14:     Obtenção dos clusters ( $c$ ) a partir das fronteiras  $f$
  - 15:     Seleção do melhor cluster ( $c_{best}$ )
  - 16:     Obtenção do conjunto de posições ( $w$ ) que ligam o drone até  $c_{best}$  utilizando Informed-RRT\*
  - 17:     Navegação do robô por todas as posições  $w$
  - 18: **Fim do Enquanto**
- 

do mapa. Por exemplo, se o robô pode explorar até uma altura de 10 metros e a resolução do OctoMap é de 0.5 metros, serão necessárias 20 imagens para representar o OctoMap ao longo do processo de exploração.

Nas linhas 3-8, descreve-se o processo que ocorre em paralelo à exploração contínua. Enquanto o robô navega, a atualização do mapa deve prosseguir para garantir que as iterações subseqüentes incorporem as alterações observadas no OctoMap durante a navegação anterior.

Nas linhas 9-18, ocorre o ciclo principal da exploração autônoma, cujo término é determinado quando um volume pré-determinado do espaço é explorado. É importante destacar que esse valor é totalmente personalizável, pois uma das premissas da exploração autônoma é a incerteza em relação ao tamanho da área, então seu valor deve ser condizente ao ambiente a ser explorado. A operação de exploração sequencial começa com a conversão do OctoMap em  $n$  imagens que representa o ambiente explorado até o momento (linha 11). Em seguida, o modelo de *deep learning* é aplicado (linha 12) para detectar as diversas fronteiras

$f$  presentes no ambiente. Essas fronteiras são avaliadas para verificar se o centro dos retângulos gerados pela rede neural está localizado em uma área conhecida e livre de obstáculos, processo este conhecido como pós-processamento (linha 13).

Com as fronteiras obtidas, é necessário clusterizá-las para eliminar as redundâncias presentes no mapa, como observado na linha 14 do Algoritmo 1. A etapa final do algoritmo consiste em determinar qual é o melhor cluster, conforme a lei de seleção apresentada neste trabalho, e encontrar um conjunto de posições que permita ao robô navegar com segurança pelo ambiente, sem obstáculos.

Após todas as posições serem percorridas, a navegação é interrompida e uma nova conversão do mapa em OctoMap é iniciada na linha 11. É importante ressaltar que algumas considerações foram incorporadas às missões, como, por exemplo, se alguma posição em que o algoritmo de geração de caminho se encontra estiver sob um obstáculo, o robô interrompe a navegação (linha 17) e reinicia o processo na linha 11.

## 5 RESULTADOS FINAIS

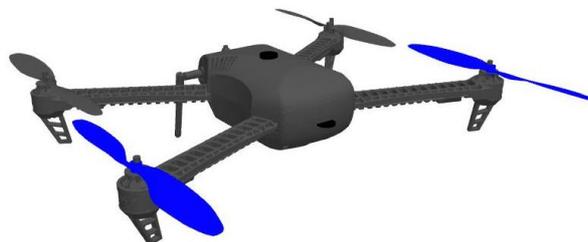
Esta seção apresenta os resultados experimentais obtidos no âmbito desta pesquisa, organizados em quatro subseções que abordam distintos cenários de validação. A primeira subseção demonstra o desempenho do sistema em ambientes *indoor* de dimensões reduzidas, característicos de aplicações em espaços confinados. A segunda subseção estende essa análise para ambientes *outdoors* de maior escala, incorporando obstáculos urbanos comuns e condições de navegação mais complexas.

O terceiro conjunto de resultados avalia a eficácia da detecção de fronteiras em dados reais coletados em campo, validando a aplicabilidade prática da metodologia proposta em situações operacionais. Por fim, a quarta subseção apresenta uma análise comparativa quantitativa que evidencia as vantagens da abordagem desenvolvida em relação ao estado da arte, destacando especificamente a característica determinística no tempo de concepção das fronteiras onde o método proposto demonstra superioridade estatisticamente significativa.

Todas os testes foram realizadas em um computador com 32GB de memória RAM e um processador AMD Ryzen 5 3600 com frequência de processamento de 3.6GHz. O sistema operacional utilizado foi o Ubuntu 20.04.6 LTS de 64 Bits. A missão autônoma foi desenvolvida utilizando a *framework* ROS (Robot Operating System) [116], cuja versão utilizada foi a Melodic. Todos os algoritmos implementados foram feitos nas linguagens de programação Python 3.8 e C++. O modelo de drone utilizado neste trabalho é o Iris [117], e uma imagem que representa o modelo pode ser observada na Figura 32.

Para a aplicação pretendida, este modelo de drone atende às necessidades, sendo capaz de transportar cargas de até 400g e oferecendo autonomia de voo entre 10 e 15 minutos. Equipado com o controlador Pixhawk [118], que utiliza os eficientes e leves protocolos MAVLink [119], o drone garante uma integração suave com o ROS. Para facilitar o acesso às informações do drone no ROS, foi utilizado o pacote MAVRos, responsável por interpretar e gerenciar todos os dados transmitidos pelo protocolo MAVLink. A simulação da placa controladora foi realizada pelo pacote PX4, que não apenas simula todos os aspectos operacionais do drone, mas também inclui o modelo Iris utilizado neste estudo.

Figura 32 – Modelo de Drone Iris.



Fonte: Retirado de [116].

Para desempenhar a missão, foram definidas as velocidades máximas horizontais de  $1m/s$  para todas condições. A velocidade vertical máxima foi de  $3m/s$ , com âmbos os valores de aceleração e desaceleração de  $3m/s^2$ . O valor máximo de detecção de obstáculos a ser considerado pela câmera foi de  $10m$ .

O sensor simulado para a detecção dos pontos 3D é o mesmo utilizado na extração dos dados que originaram o banco de dados empregado no treinamento da rede neural utilizada para a detecção de fronteiras. Essa câmera possui um sensor infravermelho capaz de identificar obstáculos no ambiente, além de um amplo campo de visão de  $87^\circ \times 58^\circ \times 95^\circ$  (vertical  $\times$  horizontal  $\times$  diagonal), conforme descrito previamente na Seção 4.1. A câmera foi posicionada de forma centralizada nos eixos  $xx$  e  $yy$  do robô, com o centro ótico deslocado verticalmente em  $50\text{ mm}$  a partir da base do robô, garantindo uma captura precisa e consistente durante a exploração.

A seleção dos parâmetros para a predição de fronteiras no modelo de *deep learning* é de extrema importância, pois impacta diretamente na qualidade das fronteiras obtidas e no tempo necessário para reconhecê-las. Uma vez definida a resolução de entrada da rede, que permanece constante para todas as predições, espera-se que o tempo de detecção das fronteiras não varie. Os parâmetros utilizados incluem uma resolução de entrada de  $240 \times 240$  pixels com um *threshold* de confiança de 5%. Este *threshold* de confiança foi configurado de forma bastante baixa,

permitindo que qualquer candidato a fronteira com uma similaridade mínima fosse reconhecido pelo modelo como uma fronteira válida. Essa abordagem foi facilitada pela simplicidade das imagens processadas (modelo proposto por este trabalho), minimizando assim possíveis classificações incorretas.

Em alguns casos, é comum haver sobreposição entre as caixas delimitadoras que compõem as fronteiras detectadas pelo modelo de rede neural. Para garantir resultados consistentes, é crucial implementar filtros para remover essas possíveis duplicidades nas fronteiras. Neste trabalho, foi utilizado o filtro de *Non-Max Suppression* cuja implementação pode ser observada em [120]. Esta metodologia atribui uma pontuação a todas as caixas delimitadoras encontradas e posteriormente remove aquelas que têm uma pontuação inferior a um limiar pré-definido. Para este estudo, o máximo de sobreposição permitido foi de 10%. Isso significa que as coordenadas que tiveram mais de 10% de sua área sobreposta por outra caixa delimitadora foram prontamente descartadas.

Por fim, ao limitar a altura máxima de navegação do robô em 10 metros, e considerando uma resolução no OctoMap de 0.5 metros, resulta em um máximo de 20 níveis no ambiente. Isso significa que serão sempre geradas 20 imagens para análise na detecção de fronteiras.

As explorações realizadas neste estudo serão comparadas ao estudo recente apresentado por Batinovic et al. [28], cujo objetivo principal é semelhante ao deste trabalho: explorar um ambiente desconhecido no menor tempo possível. Como discutido nos trabalhos correlatos, a metodologia proposta em [28] visa identificar fronteiras analisando todos os *voxels* presentes na estrutura de OctoMap, que é atualizada ao longo de toda a exploração. Todos os *voxels* são analisados e aqueles considerados fronteiras são submetidos ao algoritmo de clusterização *Mean-Shift*. Este algoritmo é interessante porque não requer uma condição inicial para a convergência da clusterização, ao contrário do K-Means. No entanto, a complexidade do *Mean-Shift* é maior que a do K-Means, o que implica em mais cálculos necessários para obter os resultados em cada detecção de fronteiras. Assim, o *Mean-Shift* leva mais tempo para realizar a clusterização de fronteiras em comparação ao K-Means.

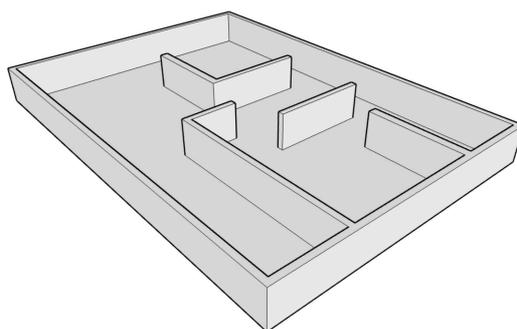
O objetivo da comparação entre esses dois algoritmos é determinar qual deles

é superior na detecção de fronteiras e também qual é o mais eficiente no processo de exploração autônoma. Para isso, serão empregados os mesmos algoritmos de planejamento de trajetória e o mesmo controlador. Embora a escolha do algoritmo de planejamento de trajetória possa influenciar na exploração, ao utilizar o mesmo planejador e controlador, a análise se concentra na eficiência da metodologia proposta neste trabalho. Para ambos os casos, será utilizado o Informed-RRT\* [24] (descrito na Seção 3.3) como o planejador de trajetória. O controlador nativo do PX4 recebe as coordenadas no espaço cartesiano com origem no ponto inicial onde o robô inicia a sua missão de exploração de modo iterativo (descrito na Seção 4.5.

### 5.1 EXPLORAÇÃO AUTÔNOMA EM AMBIENTES *INDOOR*

O primeiro resultado a ser apresentado nesta tese diz respeito à exploração de um ambiente *indoor* pelo robô. Esse ambiente é caracterizado por várias salas que servem como obstáculos, apresentando desafios para a navegação do robô. Os resultados serão avaliados por métricas de tempo e volume, medindo a rapidez com que o mapa é expandido. O objetivo principal é maximizar a expansão do mapa, reduzindo ao mínimo possível as áreas não mapeadas em um determinado tempo. As unidades de medida são segundos para o tempo e metros cúbicos ( $m^3$ ) para o volume. A Figura 33 mostra o mapa *indoor* utilizado na exploração autônoma.

Figura 33 – Mapa *Indoor* a Ser Explorado.



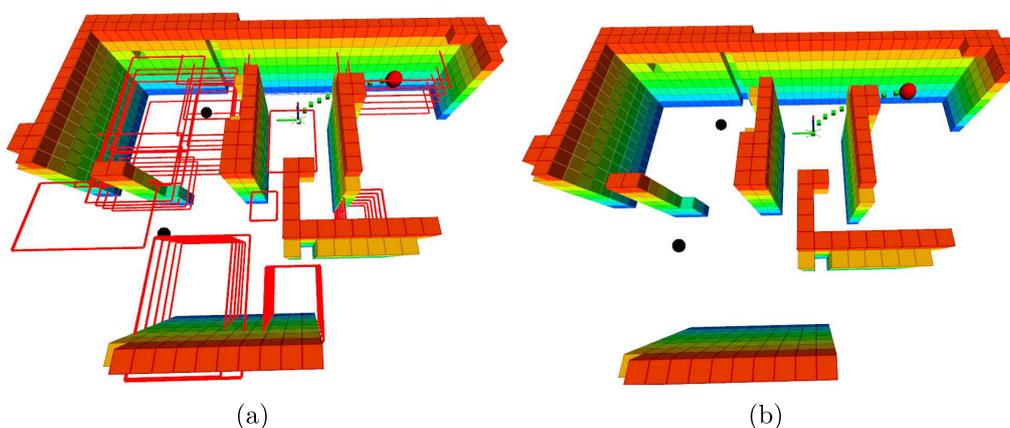
Fonte: Elaborado pelo Autor.

O mapa simulado tem dimensões de  $17m \times 12m \times 4m$ , resultando em

um volume máximo de  $816m^3$ . Devido à natureza probabilística do método de mapeamento (OctoMap) e variações nos sensores, é possível que alguns pontos no mapa não sejam completamente explorados, resultando em áreas desconhecidas que impedem o alcance total do volume máximo de  $816m^3$ . Essas pequenas áreas não exploradas são negligenciadas pelo sistema de detecção de fronteiras baseado em rede neural, pois os pixels que geram essas fronteiras são insignificantes e não afetam o volume total explorado do mapa.

A Figura 34 é gerada em um momento aleatório durante a exploração do ambiente. Na Figura 34(a), é possível notar os retângulos vermelhos que correspondem às regiões fronteiriças detectadas pelo modelo de *deep learning*. Para melhor visualização, na Figura 34(b), apresentamos a mesma imagem sem as marcações de fronteiras. Além das fronteiras, os clusters gerados por elas são representados como esferas pretas nas imagens. O caminho encontrado pelo Informed-RRT\* é demarcado por uma sequência de segmentos cúbicos verdes. Por fim, o melhor objetivo dentre aqueles disponíveis, é representado por uma esfera vermelha na Figura 34(b).

Figura 34 – Exploração *Indoor*: (a) Representação do Mapa já Explorado com Fronteira. (b) Representação do Mapa já Explorado sem Fronteira.



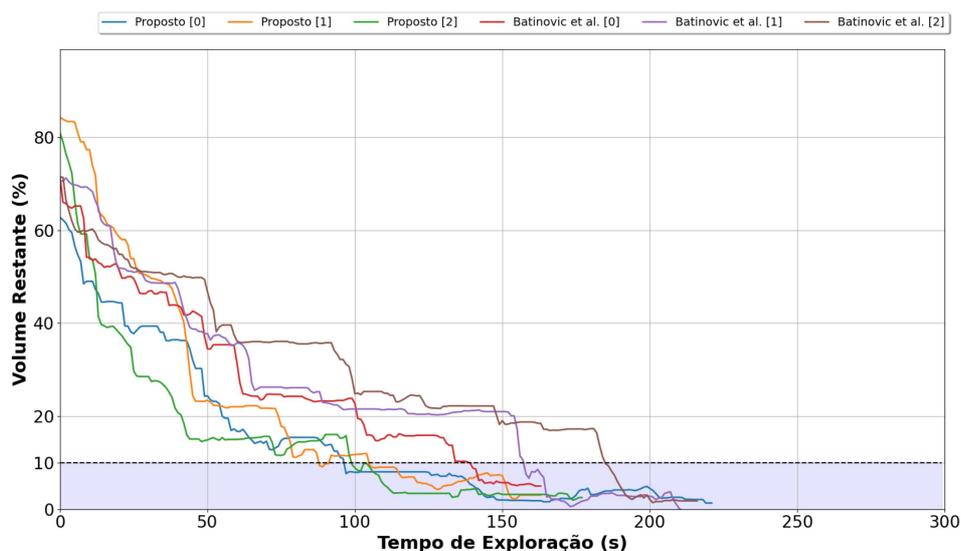
Fonte: Elaborado pelo Autor.

Para demonstrar a eficácia da metodologia proposta, as missões autônomas descritas neste estudo foram comparadas com a abordagem de exploração detalhada por Batinovic et al. [28]. O protocolo adotado especificou o uso do mesmo planejador

de trajetória em ambos os métodos, e a estrutura de mapeamento baseada em octree foi idêntica para as duas abordagens. Além disso, a política de navegação entre pontos de passagem, gerada pelo planejador de trajetória, foi consistentemente aplicada em todos os testes.

O comportamento do gráfico que mostra o volume desconhecido em relação ao tempo de exploração é considerado um indicador do desempenho das metodologias de exploração autônoma. Na Figura 35, foram comparadas três missões desenvolvidas pela metodologia proposta neste trabalho com outras três desenvolvidas em [28]. O sucesso da missão foi determinado quando 90% do volume total se tornou conhecido para ambas as metodologias, portanto, a missão será considerada concluída quando apenas 10% for desconhecido. Esse critério foi estabelecido, uma vez que o drone não possui conhecimento prévio sobre o volume total do ambiente.

Figura 35 – Comparação entre a metodologia Proposta  $\times$  Batinovic et al. [28] em Ambientes *Indoor*.



Fonte: Elaborado pelo Autor.

A Figura 35 apresenta a evolução temporal do volume desconhecido durante a fase de exploração do ambiente, partindo de um volume inicial de  $816\text{ m}^3$  e convergindo para o limiar de 10% previamente estabelecido como critério de

sucesso. A taxa de redução do volume desconhecido serve como métrica direta da eficiência exploratória, sendo que declives mais acentuados indicam maior eficácia na cobertura espacial. Nas três execuções mostradas, a metodologia proposta alcança o limiar de convergência significativamente antes da abordagem comparada, demonstrando sua superioridade qualitativa.

Para quantificar esta vantagem, realizamos 10 missões independentes com cada metodologia, calculando o tempo médio necessário para atingir o critério de 10% do volume inicial. A Tabela 2 resume os resultados desta análise comparativa:

Tabela 2 – Análise Quantitativa de Missões de Exploração Autônoma.

Metodologia	Tempo Médio (s)	Desvio Padrão (s)	Média Final Vol. (%)
<b>Proposta</b>	119.83	26.85	2.02
<b>Batinovic et al. [28]</b>	159.17	20.80	2.51

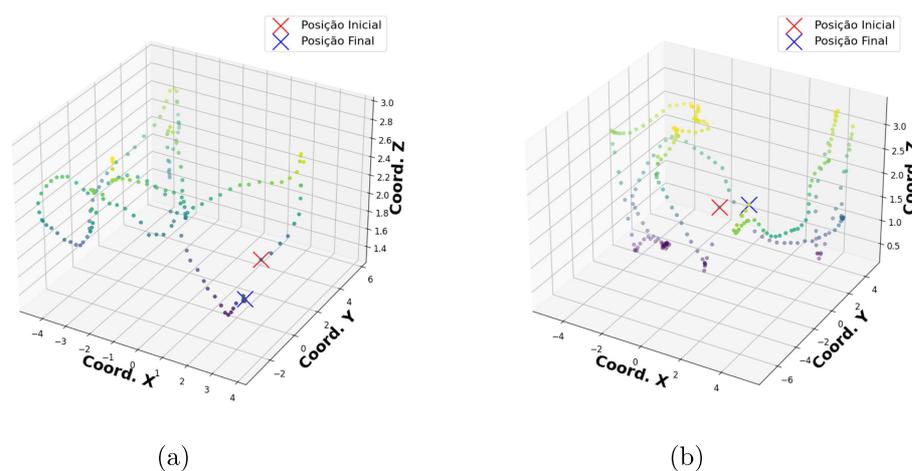
Examinando a Tabela 2, observa-se uma diferença significativa no tempo médio necessário entre as duas metodologias avaliadas. A metodologia proposta apresentou uma redução de 39.34 segundos (24.7%) em relação aos resultados obtidos por Batinovic et al. [28], destacando uma melhoria expressiva na eficiência temporal. Esta análise evidencia que a abordagem inovadora desenvolvida, com sua metodologia aprimorada de detecção de fronteiras e política de tomada de decisão inovadora, permite uma exploração mais eficiente e direcionada na busca por espaços inexplorados.

Além da vantagem temporal, os resultados demonstram que o volume residual final foi 19.5% menor na metodologia proposta (2.02% vs. 2.51%), atingindo níveis de cobertura ligeiramente superiores. Esta diferença, embora modesta, combinada com a significativa redução no tempo de exploração, reafirma a eficácia global da solução desenvolvida. O sistema mantém alta sensibilidade ao reconhecimento do espaço mapeado enquanto alcança esses resultados em um tempo substancialmente menor, estabelecendo um novo patamar de desempenho para exploração autônoma em ambientes indoor.

Outra métrica crucial na avaliação é a distância percorrida pelo robô durante a navegação. Uma metodologia eficiente não apenas minimiza o tempo de explo-

ração, mas também reduz a distância percorrida para mapear o ambiente, visto que trajetórias mais longas resultam em maior consumo energético. A Figura 36 compara visualmente os caminhos percorridos em missões típicas para ambas as metodologias no ambiente *indoor*.

Figura 36 – Caminho Percorrido no Mapa *Indoor*: (a) Metodologia Proposta (b) Metodologia de Batinovic et al. [28].



Fonte: Elaborado pelo Autor.

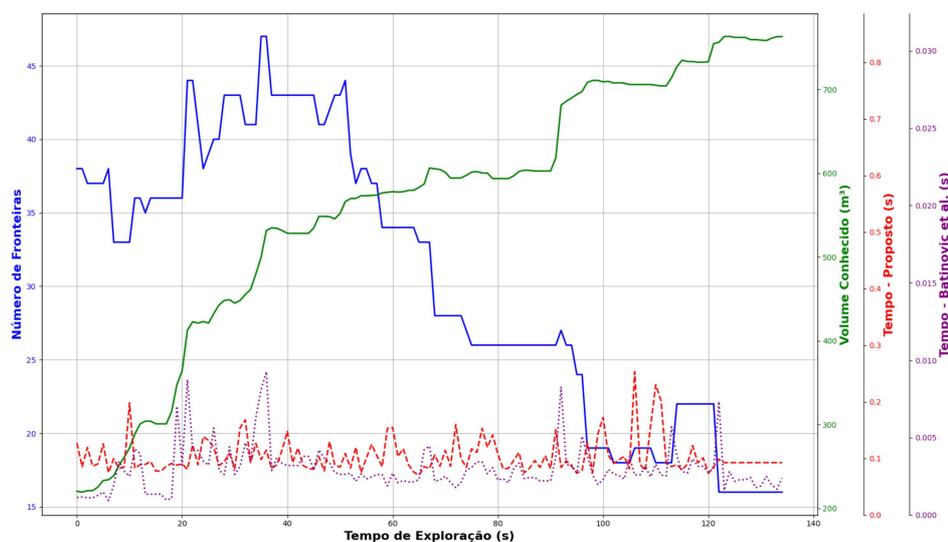
A análise qualitativa da Figura 36 revela que a metodologia proposta gera trajetórias mais diretas e com menos redundâncias, percorrendo distâncias menores para explorar 90% do ambiente. Para quantificar esta vantagem, a Tabela 3 apresenta os resultados médios de 10 execuções independentes.

Tabela 3 – Distância Total Percorrida pelo Drone ( $m$ ).

Metodologia	Distância Média	Desvio Padrão
<b>Proposta</b>	37.59	9.84
<b>Batinovic et al. [28]</b>	42.15	5.57

Os dados revelam que a abordagem proposta reduz em média 4.56 metros (10.8%) a distância total percorrida, complementando sua vantagem temporal anteriormente demonstrada. Esta dupla melhoria - em tempo de exploração e eficiência energética - consolida a superioridade da metodologia desenvolvida, que

Figura 37 – Análise de Eficiência Computacional na Detecção de Fronteiras.



Fonte: Elaborado pelo Autor.

consegue mapear ambientes complexos mais rapidamente e com menor consumo de recursos.

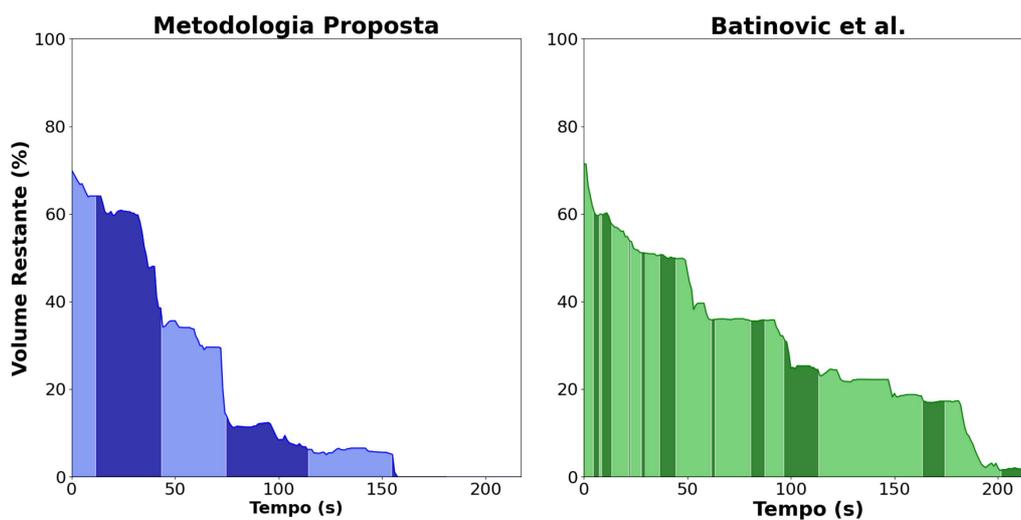
Um aspecto crítico na avaliação do desempenho é a análise da eficiência computacional no processo de detecção de fronteiras durante a exploração autônoma. A Figura 37 revela que a metodologia proposta mantém tempos de processamento consistentes (média de 0.104 segundos por detecção) mesmo com o aumento no número de fronteiras identificadas. Esta invariância temporal demonstra uma característica determinística fundamental para sistemas de exploração robustos, garantindo previsibilidade operacional independentemente da complexidade do ambiente.

Em contraste, a abordagem de Batinovic et al. [28] exibe uma correlação positiva entre o número de fronteiras e o tempo de processamento, indicando uma dependência algorítmica. Esta diferença qualitativa, inicialmente observada na análise gráfica, será quantificada e validada através de análise estatística rigorosa na Seção 5.4.1.

A eficácia da nova política de decisão pode ser quantificada analisando a

frequência de tomada de decisão necessária para completar as missões de exploração. Como evidenciado pelos tempos médios na Tabela 3, a metodologia proposta demonstra superioridade operacional em comparação com a abordagem de Batinovic et al. [28]. Esta vantagem é ainda mais pronunciada quando analisamos o número de decisões necessárias, conforme ilustrado na Figura 38.

Figura 38 – Análise Comparativa da Frequência de Tomada de Decisão.



Fonte: Elaborado pelo Autor.

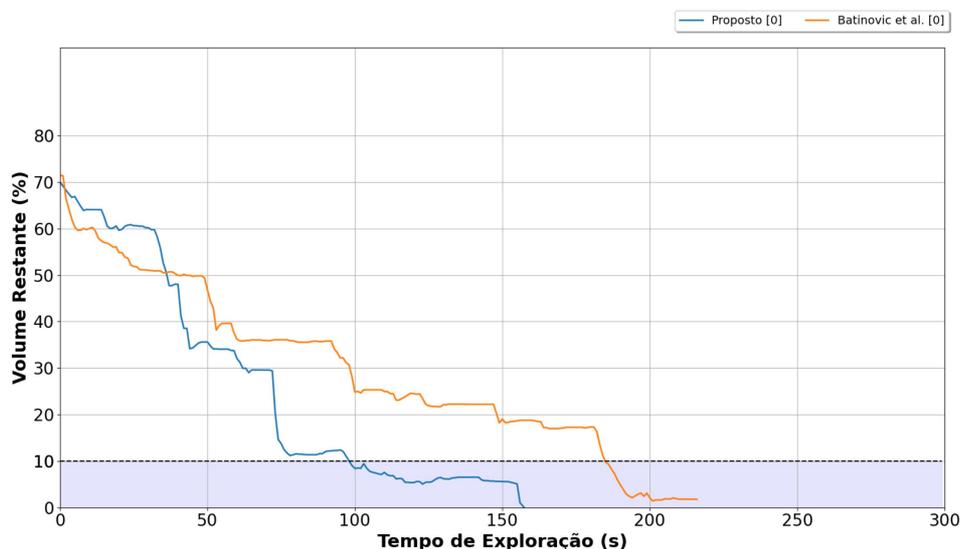
A análise da Figura 38 revela que a metodologia proposta necessita de apenas 6 eventos de decisão para concluir a exploração, em contraste com os 21 exigidos pela abordagem comparada. Cada transição entre as regiões coloridas corresponde a um novo processo decisório, demonstrando que nossa abordagem produz fronteiras mais informativas que possibilitam: seleção de *waypoints* mais estratégicos, minimização de movimentos redundantes, e expansão mais eficiente do mapa.

Essa eficiência no processo decisório é particularmente crucial para sistemas autônomos, onde decisões subótimas podem acarretar em três principais consequências negativas: aumento do consumo energético, incremento no tempo de processamento e geração de trajetórias não ideais. A combinação entre menor

frequência decisória e maior qualidade na identificação de fronteiras resulta na redução acelerada do volume não explorado, conforme observado nas análises anteriores, confirmando a superioridade da abordagem proposta tanto nos aspectos quantitativos quanto qualitativos.

Uma análise complementar da eficiência da política de tomada de decisão pode ser realizada através da inspeção da curva de volume não explorado em função do tempo de missão. Segmentos com comportamento constante nesta curva indicam períodos onde as decisões tomadas não resultaram em expansão significativa do mapa explorado. A Figura 39 apresenta uma comparação quantitativa deste comportamento entre a metodologia proposta e a abordagem de Batinovic et al. [28].

Figura 39 – Análise Comparativa da Eficiência de Exploração.



Fonte: Elaborado pelo Autor.

A análise da Figura 39 revela diferenças marcantes no comportamento exploratório. A metodologia de Batinovic et al. apresenta extensos períodos de estagnação, onde o volume não explorado permanece constante, indicando decisões subótimas que não contribuíram para a expansão do mapa. Em contraste, a metodologia proposta demonstra um decaimento quase monotônico do volume não

explorado, alcançando o limiar de 10% menos tempo. Esse padrão de decaimento contínuo evidencia que cada decisão tomada contribuiu efetivamente para a redução do espaço não mapeado, validando a superioridade da abordagem proposta em termos de eficiência exploratória.

## 5.2 EXPLORAÇÃO AUTÔNOMA EM AMBIENTES *OUTDOOR*

O segundo conjunto de resultados apresentados neste trabalho refere-se à exploração autônoma em um ambiente caracterizado como *outdoor*. Diferentemente do cenário discutido na Seção 5.1, este novo ambiente apresenta uma diversidade de elementos típicos de ambientes externos, incluindo casas, carros, pessoas, postes de iluminação e vegetação, entre outros. O objetivo principal é demonstrar que a metodologia proposta é capaz de manter o mesmo nível de desempenho quando aplicada a ambientes maiores e mais complexos. A Figura 40 ilustra o ambiente a ser explorado.

Figura 40 – Mapa *Outdoor* a Ser Explorado.



Fonte: Elaborado pelo Autor.

O mapa apresentado na Figura 40 possui dimensões de  $21 \times 21 \times 9$  metros, totalizando um volume aproximado de  $4000m^3$  a ser explorado. Devido à ampliação

do ambiente, foi adotado um espaçamento de 2 metros entre os *waypoints* gerados pelo planejador de caminhos. Essa alteração não visa reduzir o tempo necessário para obtenção dos resultados e não constitui um diferencial do algoritmo proposto, pois o mesmo espaçamento foi aplicado à metodologia de comparação apresentada por Batinovic et al. [28]. O critério para considerar a missão concluída permaneceu o mesmo utilizado no ambiente *indoor*: a missão é encerrada quando 10% do volume total do ambiente ainda restar a ser explorado, momento em que se considera que o drone atingiu seus objetivos.

Todas as diretrizes estabelecidas para a coleta de dados no ambiente *indoor* foram rigorosamente aplicadas ao cenário *outdoor*. Essa estratégia busca evidenciar a consistência e a robustez dos resultados obtidos, independentemente do tipo de ambiente explorado. Ademais, a resolução do mapa permaneceu inalterada, reafirmando a uniformidade metodológica adotada ao longo do estudo.

O tempo necessário para conclusão da exploração em ambiente externo constituiu a primeira métrica de avaliação neste estudo. Os dados quantitativos estão consolidados na Tabela 4, enquanto a Figura 41 apresenta exemplos representativos das trajetórias de exploração.

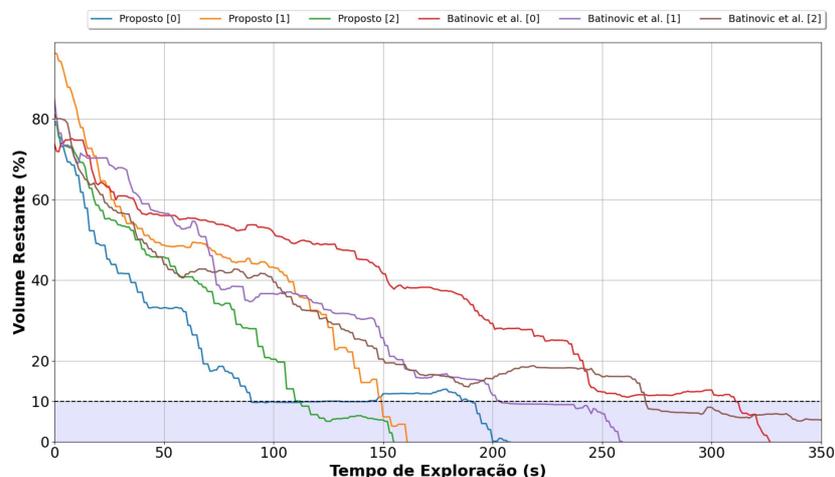
Tabela 4 – Análise Quantitativa de Missões de Exploração Autônoma.

<b>Metodologia</b>	<b>Tempo Médio (s)</b>	<b>Desvio Padrão (s)</b>	<b>Média Final Vol. (%)</b>
<b>Proposta</b>	143.25	30.02	2.53
<b>Batinovic et al. [28]</b>	290.21	43.72	4.85

Os resultados em ambiente *outdoor* corroboram as vantagens observadas nos testes *indoor*. A metodologia proposta demonstrou superioridade quantitativa, exigindo em média 146.96 segundos (50.6%) a menos que a abordagem de Batinovic et al. [28] para completar a exploração. Essa diferença acentuada evidencia a robustez da política de decisão frente à maior complexidade de ambientes abertos, combinada com a eficácia na seleção de fronteiras estratégicas e a notável adaptabilidade a variações topológicas.

Em ambientes *outdoor*, as diferenças entre as metodologias tornam-se particularmente acentuadas devido ao impacto direto das decisões de exploração na

Figura 41 – Comparação Entre a Metodologia Proposta × Batinovic et al. [28] em Ambientes *Outdoor*.



Fonte: Elaborado pelo Autor.

eficiência de mapeamento. A Tabela 5 apresenta uma análise comparativa das distâncias percorridas, revelando como decisões subótimas resultam em trajetórias significativamente mais longas, comprometendo a eficácia geral da missão.

Tabela 5 – Distância Total Percorrida Pelo Drone ( $m$ ).

Metodologia	Distância Média	Desvio Padrão
<b>Proposta</b>	75.77	15.96
<b>Batinovic et al. [28]</b>	127.90	19.33

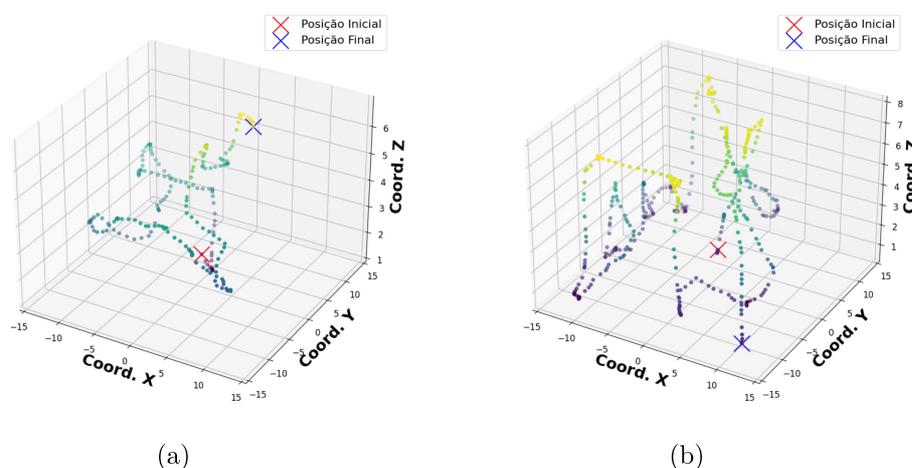
A análise dos dados revela que a metodologia proposta reduz a distância percorrida em 52.13 metros (40.8%) em relação à abordagem de Batinovic et al. [28]. Essa diferença substancial decorre diretamente da eficiência superior no processo de tomada de decisão, que resulta em seleção mais inteligente de fronteiras de exploração, minimização de movimentos redundantes e trajetórias mais diretas entre *waypoints*.

O aspecto energético constitui um parâmetro crítico em operações com drones. A redução de 40.8% na distância percorrida se traduz em economia

energética proporcional, permitindo maior autonomia operacional, cobertura de áreas mais extensas e maior margem de segurança para situações imprevistas. Esses fatores combinados estabelecem a metodologia proposta como superior para aplicações práticas em ambientes externos de grande escala.

A Figura 42 apresenta exemplos ilustrativos dos trajetos percorridos pelo drone durante as explorações realizadas no ambiente *outdoor*.

Figura 42 – Caminho Percorrido no Mapa *Outdoor*: (a) Metodologia Proposta (b) Metodologia de Batinovic et al. [28].



Fonte: Elaborado pelo Autor.

A análise da Figura 42 corrobora os resultados obtidos para a distância média percorrida, demonstrando claramente a superioridade da metodologia proposta em relação à abordagem convencional. A Figura 42(b) revela uma quantidade significativamente maior de movimentos durante a exploração, acompanhada por variações mais pronunciadas na altura de voo (eixo Z). Essas oscilações verticais, embora indiquem uma exploração ativa do espaço tridimensional, sugerem trajetórias menos otimizadas na abordagem comparada.

Em contraste com os ambientes *indoor*, onde a altura de voo é naturalmente restrita, o cenário *outdoor* evidencia diferenças marcantes nas estratégias de navegação. Para quantificar este comportamento, realizamos uma análise específica das variações de altura, considerando os valores absolutos das mudanças consecutivas

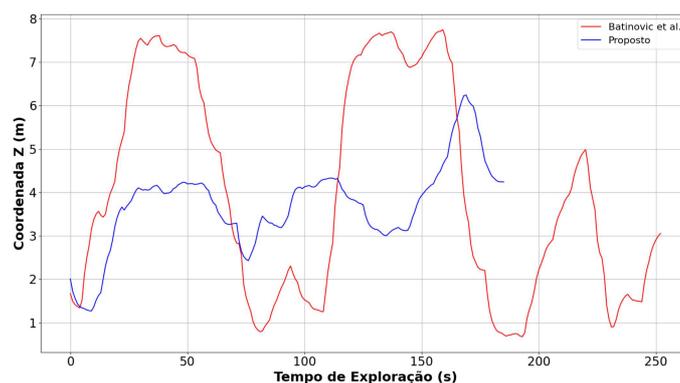
na coordenada vertical. Os resultados, apresentados na Tabela 6, revelam padrões distintos entre as metodologias.

Tabela 6 – Análise Quantitativa da Variação de Altura em Navegação *Outdoor*.

Metodologia	Varição Média (m) (s)	Desvio Padrão (s)
<b>Proposta</b>	21.46	4.88
<b>Batinovic et al. [28]</b>	68.86	20.69

Os dados demonstram que a abordagem de Batinovic et al. apresenta variações de altura 3.2 vezes maiores que a metodologia proposta (68.86m vs. 21.46m). Esta diferença acentuada pode ser atribuída à estratégia inovadora de clusterização implementada em nosso trabalho, que agrega múltiplas áreas detectadas pela rede neural em centroides estratégicos. Essa abordagem possibilita seleção mais inteligente de *waypoints* tridimensionais, minimização de movimentos verticais desnecessários e trajetórias mais suaves e energeticamente eficientes, resultando em uma exploração mais sistemática e com menor consumo energético. A Figura 43 ilustra as curvas correspondentes à coordenada Z para as duas metodologias em comparação.

Figura 43 – Comparação Entre Metodologias com Base na Variação de Altitude Durante a Exploração.

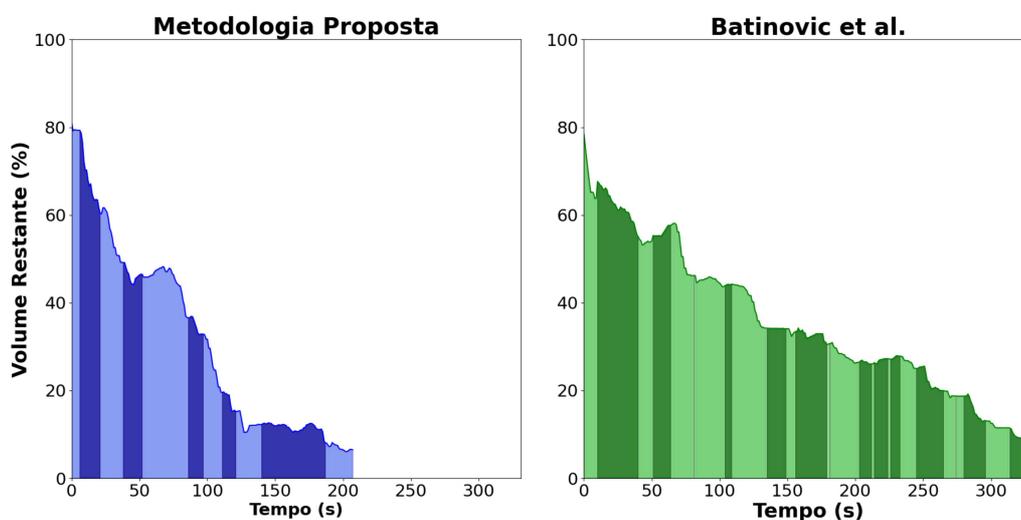


Fonte: Elaborado pelo Autor.

A análise revela que a metodologia comparada requer variações significativas de altitude para alcançar a exploração completa do ambiente. Esta abordagem, baseada na exploração exaustiva de *grids* individuais em cada nível de altura, apresenta um padrão operacional cíclico que primeiro esgota todas as fronteiras no nível atual antes de realizar movimentações verticais para níveis adjacentes, repetindo este processo até a conclusão da missão. Tal estratégia resulta em trajetórias energeticamente ineficientes e tempo total de missão prolongado.

Em contraste, a metodologia proposta demonstra características operacionais marcadamente superiores. A integração entre a detecção neural de fronteiras e o algoritmo de clusterização gera uma política de decisão mais inteligente, resultando em redução de movimentos verticais desnecessários, seleção otimizada de waypoints estratégicos e trajetórias mais suaves e eficientes. Esses avanços são particularmente evidentes na Figura 44, que compara visualmente os processos de tomada de decisão em ambientes *outdoor*, demonstrando como a abordagem proposta minimiza transições entre níveis de altitude enquanto maximiza a eficiência exploratória.

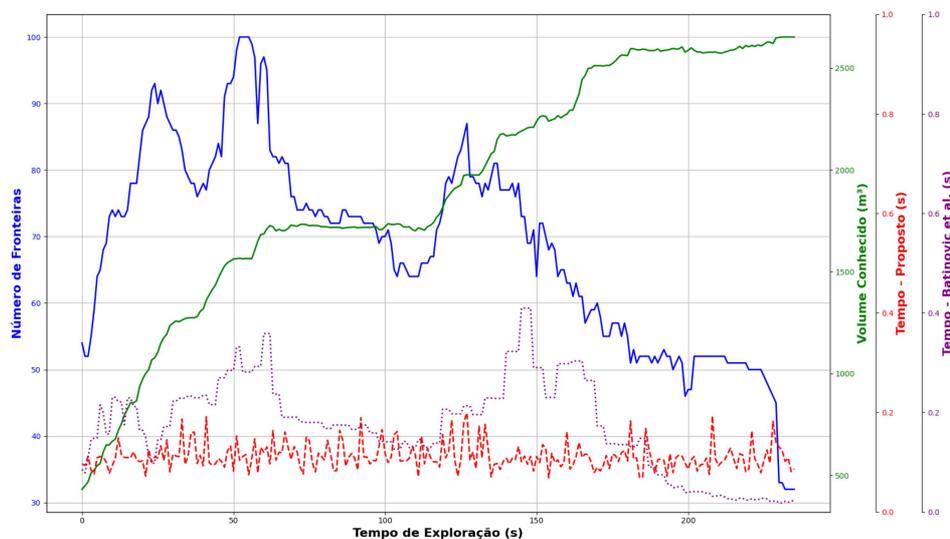
Figura 44 – Análise Comparativa da Frequência de Tomada de Decisão.



Fonte: Elaborado pelo Autor.

A análise comparativa revelou uma diferença marcante na frequência de

Figura 45 – Análise de Eficiência Computacional na Detecção de Fronteiras.



Fonte: Elaborado pelo Autor.

tomada de decisão entre as metodologias avaliadas. Enquanto a abordagem convencional exigiu 32 eventos decisórios para completar a exploração, a metodologia proposta alcançou o critério de 10% de área não explorada com apenas 12 decisões, representando uma redução de 62.5% na complexidade do processo decisório. Este resultado quantitativo corrobora a superioridade da abordagem desenvolvida.

Como avaliação final, analisamos o desempenho temporal da metodologia proposta na detecção de fronteiras em ambientes *outdoor*. Em cenários de grande escala, onde o número de fronteiras tende a ser significativamente maior, a abordagem desenvolvida demonstrou robustez computacional, superando as limitações das metodologias convencionais que apresentam degradação de desempenho proporcional ao aumento da complexidade do ambiente. A Figura 45 ilustra este comportamento, confirmando a eficiência da solução proposta mesmo em condições desafiadoras.

A análise revela que a metodologia proposta mantém um tempo médio constante de aproximadamente 0.110 segundos por detecção, independentemente do número de fronteiras presentes - característica determinística essencial para

sistemas de exploração autônoma. Em contraste, a abordagem de Batinovic et al. [28] exibiu a mesma dependência linear observada em ambientes *indoor*, com tempo de processamento crescendo proporcionalmente à quantidade de fronteiras.

Comparando os ambientes *indoor* e *outdoor*, observamos que enquanto o número máximo de fronteiras aumentou de 47 para 100 (aumento de 112.8%), o tempo de processamento da metodologia proposta permaneceu estável (variação inferior a 5.8% entre as médias de 0.104s e 0.110s). Por outro lado, a abordagem comparada sofreu um aumento de 5666.7% no tempo médio de detecção (de 0.003s para 0.173s), evidenciando sua sensibilidade à escala do ambiente.

Estes resultados demonstram claramente a vantagem da metodologia proposta em cenários complexos: sua arquitetura computacional eficiente, baseada na detecção de fronteiras através de uma regressão de redes neurais, mantém o desempenho estável independentemente da complexidade do ambiente. A análise estatística completa, incluindo os resultados médios de todas as missões realizadas, será apresentada na Seção 5.4.1, fornecendo uma avaliação mais abrangente do desempenho comparativo.

### 5.3 DETECÇÃO DE FRONTEIRAS EM UM AMBIENTE REAL

Para validar a robustez e eficiência da metodologia proposta em condições reais, aplicamos nosso algoritmo de detecção de fronteiras a um banco de dados público amplamente reconhecido na literatura [121]. Este conjunto de dados foi gerado no âmbito do projeto *AeroARMs: Aerial Robots with Advanced Manipulation Capabilities for Inspection and Maintenance*, desenvolvido por um consórcio internacional de universidades europeias [122].

O projeto AeroARMs visa integrar manipuladores robóticos a plataformas aéreas não tripuladas, permitindo a execução de tarefas complexas de inspeção e manutenção em infraestruturas críticas de difícil acesso. O principal desafio tecnológico abordado pelo projeto consiste em manter a estabilidade do veículo durante operações que requerem contato físico. Para superar este desafio, o sistema incorpora uma arquitetura sensorial avançada composta por Lidar 3D para modelagem geométrica do ambiente, câmeras estereoscópicas para percepção

visual, GPS RTK para posicionamento de alta precisão, e sensores de fluxo óptico combinados com IMUs para estimativa de movimento precisa.

Foram analisadas duas missões de inspeção teleoperadas, nas quais o drone executava um trajeto desde uma posição inicial até tubulações industriais para realização de inspeções de contato, retornando posteriormente ao ponto de origem. Os dados, disponibilizados no formato *rosbags*, permitiram a reconstrução fiel das condições operacionais reais para avaliação de nossa metodologia. A Figura 46 apresenta o sistema durante uma inspeção em tubulações de petróleo e gás, ilustrando o cenário de aplicação real considerado em nossos testes experimentais.

Figura 46 – Tarefa de Inspeccionamento via Contato Físico pelo Drone do Projeto AeroARMs.



Fonte: Retirado de [122].

Os testes foram conduzidos em um ambiente externo à universidade para que fossem considerados fatores naturais, como iluminação e ventos, que poderiam impactar a missão de inspeção do robô. A área exata do ambiente não foi especificada pelos autores do trabalho, uma vez que essa informação não era relevante no contexto da inspeção das tubulações. No entanto, este estudo considerou uma área mapeada com dimensões de  $30\text{ m} \times 30\text{ m} \times 9\text{ m}$ .

Entre os instrumentos mencionados, apenas o Lidar 3D e o GPS desempenham papéis cruciais para a exploração autônoma desta tese. O Lidar será o responsável pela geração do mapa tridimensional necessário para a detecção de fronteiras, enquanto o GPS assegura a localização precisa do robô durante todo o processo de navegação. O modelo do Lidar utilizado, o Velodyne HDL-32, destaca-se por seu amplo campo de visão horizontal de  $360^\circ$  e vertical de  $40^\circ$  ( $-10^\circ$  a  $30^\circ$ ), além de oferecer um alcance máximo de detecção de 80 a 100 metros [123]. Essas características tornam o equipamento ideal para aplicações que demandam mapeamento preciso e completo de ambientes complexos.

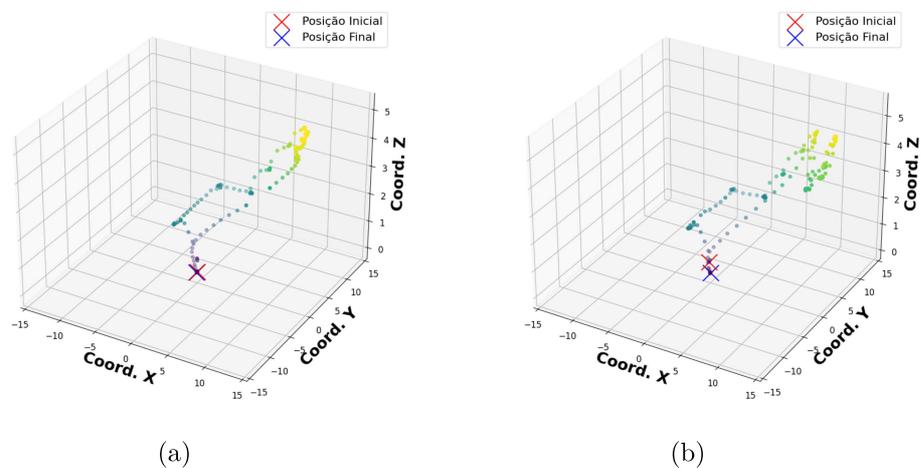
A avaliação será focada exclusivamente na capacidade de extração de fronteiras, uma vez que a tomada de decisão não se aplica neste caso, dado que os *rosbags* correspondem a eventos já ocorridos. Entretanto, a melhor fronteira será continuamente calculada, assim como o planejador de caminho que conecta a posição atual do robô à fronteira selecionada. Dessa forma, a metodologia proposta será utilizada em sua totalidade, garantindo que não haja perdas na qualidade das análises realizadas.

A primeiro resultado obtido através aplicação da metodologia proposta nos *rosbags* são os gráficos correspondentes à sua movimentação. Os resultados para as duas missões estão presentes na Figura 47.

Observa-se que a inspeção realizada na segunda missão foi mais detalhada, como evidenciado por uma maior quantidade de movimentos na parte superior do ambiente, destacados pelos pontos amarelos na Figura 47. As distâncias computadas foram de  $57.7m$  na primeira missão contra  $76.5m$  na segunda. Essa característica contribuiu para o maior tempo de execução na Missão 2, que totalizou aproximadamente 188 segundos, em comparação aos 166 segundos registrados na primeira missão.

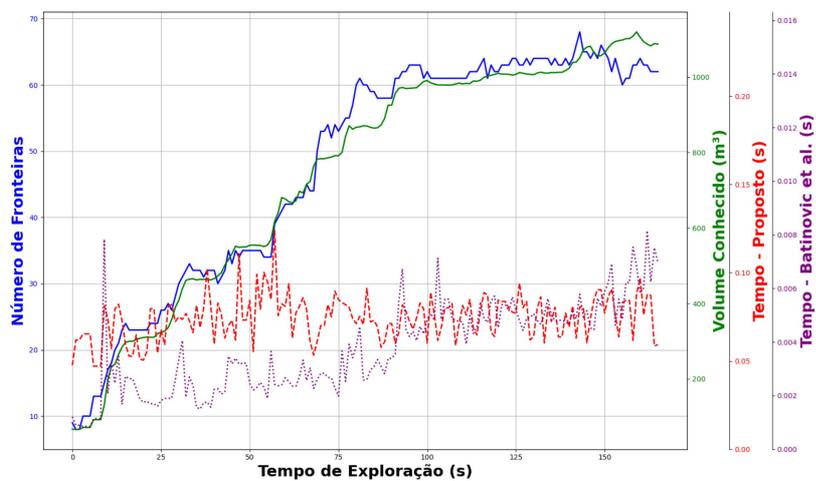
O tempo de detecção de fronteiras pode ser avaliado nos gráficos que comparam o volume conhecido e o número de fronteiras. O gráfico para as duas missões poderão ser vistos nas Figuras 48 e 49.

Figura 47 – Caminho Percorrido no Ambiente Real: (a) Missão 1 (b) Missão 2.



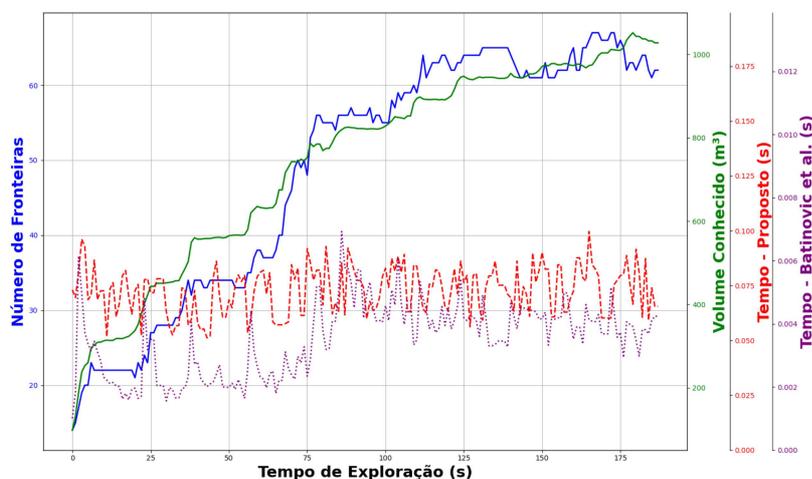
Fonte: Elaborado pelo Autor.

Figura 48 – Análise de Eficiência Computacional na Detecção de Fronteiras para a Missão 1.



Fonte: Elaborado pelo Autor.

Figura 49 – Análise de Eficiência Computacional na Detecção de Fronteiras para a Missão 2.



Fonte: Elaborado pelo Autor.

A partir das Figuras 48 e 49, é possível observar que ambos os métodos analisaram as fronteiras em tempos reduzidos, sem impactar significativamente o processo de reconhecimento das fronteiras ao longo da exploração. A metodologia proposta apresentou tempos médios ligeiramente superiores (0.074s e 0.073s) em comparação com a metodologia de Batinovic et al. [28] que apresentou 0.003s para ambas as missões. No entanto, esses tempos menores não se traduzem em um resultado mais eficiente na exploração autônoma, pois a maior parte do tempo da missão é consumida pelo deslocamento do robô durante a exploração. Nesse contexto, decisões mais precisas têm um impacto mais significativo na eficiência global do processo do que a redução no tempo de detecção das fronteiras.

Os valores de 0.003 segundos, observados na exploração prática, permaneceram próximos aos registrados no ambiente de menor complexidade (*indoor*) e distante dos valores observados no ambiente *outdoor*. Essa característica deve-se ao fato de que as fronteiras não estavam sendo efetivamente exploradas. Em outras palavras, o deslocamento do robô não era direcionado para áreas com maior potencial de abertura do mapa, mas sim para locais específicos onde as tarefas

de inspeção deveriam ser realizadas (missão teleoperada). Conseqüentemente, o processo de assimilação de novas fronteiras ocorria com pouca frequência, o que fazia com que as fronteiras detectadas permanecessem praticamente estáticas ao longo de toda a missão. Assim, uma vez identificadas, não havia necessidade de reavaliá-las para inclusão no conjunto de fronteiras, o qual, por sua vez, não era explorado.

Outro ponto relevante observado nas Figuras 48 e 49 refere-se à característica não determinística da metodologia desenvolvida por Batinovic et al. [28]. Nota-se que o tempo necessário para a detecção das fronteiras aumenta à medida que o número de fronteiras do ambiente também cresce. Como mencionado ao longo deste trabalho, essa característica é pouco atrativa para aplicações de exploração autônoma em ambientes desconhecidos, uma vez que o volume conhecido e a quantidade de fronteiras disponíveis não são informações previamente conhecidas, podendo comprometer a robustez e a adaptabilidade da exploração. Esta correlação será quantificada e aprofundada na seção 5.4.

#### 5.4 ANÁLISE DO DESEMPENHO PARA A EXTRAÇÃO DE FRONTEIRAS

Na seção de resultados relacionados ao ambiente *indoor* e *outdoor*, foi apresentado o tempo médio necessário para o robô realizar completamente a missão de exploração. Esta métrica é crucial para avaliar em sua totalidade o método na solução do problema da exploração de ambientes desconhecidos. No entanto, é igualmente importante avaliar o desempenho na identificação das fronteiras pelo drone. Conforme detalhado na metodologia deste trabalho, o processo de obtenção de fronteiras envolve quatro fases: conversão do OctoMap em imagens, inferência sobre as imagens, pós-processamento das fronteiras e clusterização.

É evidente que nossa metodologia segue um caráter determinístico para resolver o problema, uma vez que a quantidade de imagens utilizadas no processo é fixa ao longo de toda a operação. Além disso, características como a resolução da imagem permanecem inalteradas durante todo o processo. Esta característica é uma das principais motivações deste trabalho, pois em ambientes desconhecidos, a quantidade de fronteiras é totalmente variável, tornando uma metodologia independente dessa quantidade extremamente valiosa. Entre as fases mencionadas

anteriormente no processo de obtenção de clusters, apenas o algoritmo de clusterização é afetado pela complexidade da quantidade de fronteiras; portanto, quanto mais fronteiras, mais tempo é necessário para a clusterização. No entanto, essa variação foi mínima e não afetou significativamente a metodologia proposta. Esta constatação é corroborada pela Figura 37 e 45, onde é perceptível que, mesmo com um aumento considerável no número de fronteiras, o tempo necessário para obtê-las permanece praticamente constante.

Conforme explanado na seção que detalha o funcionamento do modelo de rede neural utilizado neste trabalho para a detecção de fronteiras em imagens, observa-se que o problema se reduz a uma questão de regressão. Dessa forma, ao submeter as imagens à rede, a detecção de fronteiras ocorre de maneira consistente, independentemente da quantidade de fronteiras presente nas imagens. Para ilustrar este resultado, examinamos a detecção de fronteiras em duas imagens distintas. A primeira representa uma área já explorada pelo robô, contendo múltiplas fronteiras a serem identificadas. Já a segunda imagem representa uma região não explorada, onde nenhuma fronteira é esperada. Esses cenários estão exemplificados na Figura 50. Para cada teste, foram realizadas 1000 amostragens temporais, cujos resultados estão sumarizados na Tabela 7.

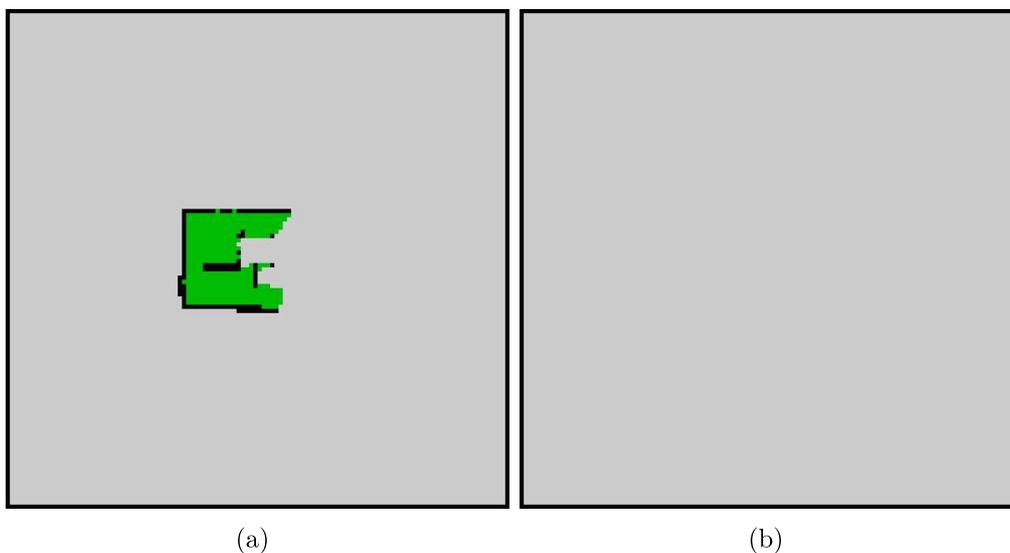
Tabela 7 – Análise Comparativa da Detecção de Fronteiras em uma Imagem.

<b>Análise</b>	<b>Tempo Médio (ms)</b>	<b>Fronteiras</b>	<b>Clusters</b>
<b>Com Fronteiras</b>	8.2 (1.8)	10.99 (0.21)	3
<b>Sem Fronteiras</b>	7.1 (1.8)	0	0

Conforme esperado, os resultados observados na Tabela 7 revelam tempos muito próximos para a detecção de fronteiras em cenários distintos: um com aproximadamente 11 fronteiras e outro sem nenhuma fronteira detectada. Os valores de desvio padrão para cada medida estão entre parênteses ao lado dos valores médios. Este resultado corresponde à apenas uma fase da metodologia proposta, desta maneira, o tempo necessário para processar a melhor fronteira na qual o robô deve se deslocar, considerando as quatro fases da metodologia proposta, é detalhado na Tabela 8.

A etapa de conversão do OctoMap em imagens ocorrem simultaneamente na

Figura 50 – (a) Cenário 1 - Com Fronteiras (b) Cenário 1 - Sem Fronteiras.



Fonte: Elaborado pelo Autor.

Tabela 8 – Tempo Necessário para Detectar Todas as Fronteiras (ms).

Fase da Metodologia	Tempo Médio	Desvio Padrão
1- OctoMap Para Imagens	8.6	4.6
2- Detecção de Fronteiras (20 Imagens)	39.7	12.1
3- Pós-Processamento	19.7	7.2
4- Clusterização	17.4	11.0

metodologia principal, sem impactar diretamente no tempo da detecção de fronteiras. No entanto, as fases subsequentes, detecção de fronteiras, pós-processamento e clusterização, são realizadas sequencialmente, e a soma dos tempos necessários reflete o desempenho global de nossa abordagem. É importante destacar que o tempo necessário para detectar fronteiras em 20 imagens não equivale ao tempo de análise de uma única imagem multiplicado por 20, como evidenciado na Tabela 7. A estrutura da YOLO é capaz de otimizar o processamento em lote, resultando em tempos inferiores quando as imagens são enviadas em conjunto.

Vale ressaltar que a necessidade de detecção de fronteiras surge esporadicamente durante a exploração autônoma, sendo necessário apenas quando o robô está parado e precisa determinar seu próximo deslocamento. Portanto, podemos

afirmar que a detecção de fronteiras não é um requisito contínuo durante todo o processo de exploração. No entanto, o mapeamento deve ocorrer continuamente ao longo do processo exploratório.

Embora o tempo de detecção de fronteiras não seja utilizado durante toda a missão e não constitua, diretamente, um indicador do aumento no tempo da exploração autônoma, é relevante demonstrar a dependência entre o número de fronteiras e o tempo necessário para detectá-las. Essa análise será realizada por meio de estatísticas descritivas, conforme apresentado na Seção 5.4.1.

#### 5.4.1 Análise Estatística da Correlação Entre o Tempo de Detecção de Fronteiras e o Volume Conhecido

Um dos principais objetivos deste trabalho é propor uma metodologia capaz de reconhecer fronteiras de forma que o tempo necessário para essa detecção seja previsível, ou seja, independentemente da quantidade de fronteiras presentes, o tempo de detecção deve apresentar variações mínimas. Embora os resultados apresentados até o momento por esta tese tenham demonstrado esse comportamento por meio de representações gráficas, é essencial complementar essa análise com uma abordagem quantitativa fundamentada em estatística descritiva, a fim de consolidar os benefícios da metodologia proposta.

Uma das formas mais eficazes de avaliar esses resultados é por meio da análise da correlação entre os dados coletados ao longo do processo. Dois conjuntos de dados que possuem uma forte correlação apresentam uma relação em que a variabilidade de um conjunto pode ser, em grande parte, explicada pelo outro. Na literatura, diversos métodos são empregados para analisar essa correlação, cada um direcionado a diferentes tipos de relações entre os sinais. Neste trabalho, serão utilizados dois métodos complementares: a correlação de Pearson [124], que mede a relação de linearidade entre os dados, e a correlação de Spearman [125], que avalia relações monotônicas, independentemente de sua linearidade.

A correlação de Pearson avalia a linearidade entre dois conjuntos de dados, quantificando a força e a direção dessa relação linear. Esse coeficiente é insensível a escalas e unidades, o que permite analisar comparações entre variáveis distintas, como o número de fronteiras detectadas e o tempo necessário para essa detecção.

O coeficiente de Pearson varia entre -1 e 1, onde:

- -1 indica uma forte correlação linear negativa, ou seja, à medida que uma variável aumenta, a outra diminui na mesma proporção;
- 1 indica uma forte correlação linear positiva, ou seja, quando uma variável aumenta, a outra também aumenta proporcionalmente;
- 0 significa que não há relação linear detectável entre as variáveis analisadas.

Essa característica torna a correlação de Pearson uma ferramenta indispensável para avaliar relações lineares entre conjuntos de dados, especialmente em estudos envolvendo variáveis de diferentes naturezas. No contexto das duas missões autônomas práticas analisadas neste trabalho, foram calculados os coeficientes de correlação entre o número de fronteiras e os respectivos tempos de detecção de fronteiras. A metodologia desenvolvida por Batinovic et al. [28] apresentou um coeficiente médio de Pearson de 0.917, indicando uma forte relação linear positiva entre essas variáveis. Por outro lado, a metodologia proposta, baseada em rede neural, apresentou um coeficiente médio de 0.233, evidenciando uma correlação significativamente menor.

Esses resultados demonstram que a metodologia presente na literatura apresenta uma maior dependência linear entre o número de fronteiras e o tempo de detecção, enquanto a abordagem baseada em rede neural é menos influenciada por essa variável. Tal característica reforça o argumento de que a metodologia proposta é mais robusta, pois opera de maneira mais consistente independentemente do volume de fronteiras detectadas.

A correlação de Spearman é especialmente valiosa para avaliar relações que não seguem um comportamento linear, destacando-se por sua aplicabilidade em contextos onde as variáveis apresentam comportamentos mais complexos. Ao analisar o comportamento do número de fronteiras detectadas ao longo da tarefa de exploração, observa-se que a quantidade de fronteiras varia de forma não linear, aumentando e diminuindo à medida que o drone explora o ambiente. Essa característica pode prejudicar a análise pela correlação de Pearson, projetada para avaliar exclusivamente relações lineares. Por essa razão, o cálculo da correlação de

Spearman surge como uma alternativa mais adequada, permitindo uma avaliação mais robusta e abrangente por meio de estatísticas descritivas, especialmente em cenários que envolvem relações de monotonicidade entre as variáveis.

A correlação de Spearman mede a relação de monotonicidade entre duas variáveis. Isso significa que, se uma variável aumenta, a outra também aumenta ou permanece constante, e, se uma variável diminui, a outra diminui ou permanece constante. Diferentemente da correlação de Pearson, a correlação de Spearman não avalia a linearidade entre as variáveis, mas sim a tendência de associação monotônica.

O cálculo da correlação de Spearman é realizado por meio da ordenação dos dados em postos. Os valores das variáveis são organizados em ordem crescente e substituídos por suas respectivas posições nos postos. Em seguida, as posições são comparadas entre as variáveis. É importante destacar que os valores absolutos das amostras não são considerados; apenas suas posições relativas nos postos são analisadas. Os valores da correlação de Spearman variam de -1 a 1, e suas interpretações são as seguintes:

- -1 indica uma forte correlação monotônica negativa, ou seja, à medida que uma variável aumenta, a outra diminui, sem necessariamente obedecer a uma relação linear;
- 1 indica uma forte correlação monotônica positiva, ou seja, à medida que uma variável aumenta, a outra também aumenta, sem necessariamente obedecer a uma relação linear;
- 0 indica ausência de relação de monotonicidade entre as variáveis.

Os cálculos da correlação de Spearman indicaram uma forte relação monotônica entre o número de fronteiras e o tempo de detecção de fronteiras na metodologia proposta por Batinovic et al. [28], com um valor de 0.901. Em contraste, para a metodologia baseada em rede neural, o valor encontrado foi de 0.221, caracterizando uma relação monotônica fraca. Esse resultado reforça, mais uma vez, o desempenho superior da implementação da detecção de fronteiras por meio da regressão, evidenciado pela menor correlação observada.

A análise das correlações também são factíveis de serem aplicadas nos resultados simulados deste trabalho. Todos os resultados, inclusive os práticos já apresentados nesta seção estão condensados na Tabela 9.

Tabela 9 – Correlação Entre os Tempos de Detecção de Fronteiras e a Quantidade de Fronteiras.

Resultados	Proposto		Batinovic et al. [28]	
	Pearson	Spearman	Pearson	Spearman
Ambiente <i>Indoor</i>	0.332	0.313	0.742	0.674
Ambiente <i>Outdoor</i>	0.121	0.137	0.951	0.980
Teste Prático	0.233	0.221	0.917	0.901

Os resultados apresentados na tabela corroboram as análises gráficas discutidas anteriormente. A metodologia desenvolvida por Batinovic et al.[28] demonstra uma forte dependência em relação ao número de fronteiras em todos os cenários analisados, enquanto a metodologia proposta apresenta uma correlação significativamente mais fraca. É importante notar que as definições sobre o que constitui uma correlação fraca são, em grande parte, empíricas e devem ser interpretadas conforme o contexto e os objetivos específicos do estudo. Para fins práticos, uma classificação amplamente aceita na comunidade acadêmica é fornecida por Evans [126], cujas definições estão detalhadas na Tabela 10.

Tabela 10 – Classificação de Correlações Segundo Evans [126].

Coefficiente (módulo)	Interpretação
0.00 - 0.19	Muito fraca
0.20 - 0.39	Fraca
0.40 - 0.59	Moderada
0.60 - 0.79	Forte
0.80 - 1.00	Muito forte

Em todos os resultados analisados, a correlação obtida para a metodologia proposta foi classificada como "Muito Fraca" ou "Fraca", enquanto a metodologia de comparação apresentou correlações classificadas como "Forte" e "Muito Forte". Um ponto interessante a ser destacado é que a classificação "Fraca" foi encontrada nos ambientes *indoor*, onde a exposição às fronteiras é naturalmente menor. Essa menor quantidade de fronteiras pode introduzir certa variabilidade nos resultados,

afetando a consistência da correlação. No entanto, essa variação não foi suficiente para comprometer os resultados obtidos, que permanecem alinhados aos objetivos e evidenciam a robustez da metodologia proposta.

Por fim, a Tabela 11 apresenta os valores médios dos tempos de detecção de fronteiras obtidos nos experimentos simulados realizados neste trabalho. A análise da tabela evidencia que a metodologia proposta por Batinovic et al. [28] enfrenta dificuldades em tarefas de exploração autônoma em ambientes com longas exposições a conjuntos extensos de fronteiras, resultando em um aumento considerável no tempo médio de detecção.

Tabela 11 – Tempos Médios para a Detecção de Fronteiras.

<b>Ambiente</b>	<b>Proposto</b>		<b>Batinovic et al. [28]</b>	
	<b>T. Médio (s)</b>	<b>D. Padrão</b>	<b>T. Médio (s)</b>	<b>D. Padrão</b>
<b><i>Indoor</i></b>	0.101	0.015	0.001	0.001
<b><i>Outdoor</i></b>	0.123	0.049	0.107	0.063

Analisando a Tabela 11, observa-se um aumento expressivo no tempo médio da metodologia de comparação. Enquanto a metodologia proposta neste trabalho apresentou um acréscimo modesto de aproximadamente 20% no tempo médio, a metodologia de comparação evidenciou um aumento extraordinário de cerca de 10600%.

## 6 CONCLUSÕES

Esta pesquisa apresentou uma nova metodologia para exploração e mapeamento autônomo de terrenos desconhecidos utilizando UAVs. No contexto desta tese de doutorado, foram propostas duas contribuições significativas para a comunidade acadêmica, visando aprimorar a capacidade do robô na exploração autônoma do ambiente.

A primeira contribuição consiste na aplicação de técnicas de *deep learning* para a detecção de fronteiras, que orientam o deslocamento do robô na expansão do mapa. A incorporação de redes neurais à solução deste problema possibilitou uma detecção dessas regiões de forma determinística. Independentemente da quantidade de fronteiras presentes no ambiente, a metodologia proposta mantém um tempo praticamente constante para a identificação das regiões fronteiriças, uma característica fundamental para cenários em que a quantidade de fronteiras é desconhecida a priori.

A segunda contribuição deste trabalho consiste na introdução de uma nova política de tomada de decisão para definir qual fronteira o robô deve explorar. Essa abordagem inovadora combina as informações extraídas pela rede neural com o custo calculado por um planejador de caminhos, atribuindo valores aos clusters gerados na detecção de fronteiras.

O planejador utilizado na política de tomada de decisão é o Direct-DRRT\*, uma metodologia previamente aplicada em robôs móveis e manipuladores robóticos, reconhecida por sua eficiência na identificação rápida de caminhos viáveis entre um ponto inicial e um ponto objetivo. Essa escolha garante que as decisões do robô sejam não apenas precisas, mas também otimizadas em termos de tempo e viabilidade de navegação no ambiente explorado.

Os resultados deste trabalho foram obtidos por meio de simulações e validados em um experimento prático. A simulação foi conduzida em dois cenários distintos: um ambiente *indoor* e outro *outdoor*. A principal diferença entre esses ambientes reside no volume a ser explorado e na distribuição de objetos no mapa, o que influencia diretamente a complexidade da navegação.

Todos os resultados foram comparados à metodologia desenvolvida por

Batinovic et al. [28], utilizando um conjunto de métricas e análises para avaliação do desempenho. Entre os critérios analisados estão o tempo necessário para expandir o mapa conhecido, a distância percorrida pelo robô, a quantidade de decisões tomadas ao longo da missão e a correlação entre os tempos de detecção das fronteiras e o número de fronteiras encontradas. Essas análises permitiram quantificar a eficiência e a robustez da abordagem proposta, destacando suas vantagens em relação à metodologia de referência.

O objetivo principal foi alcançado em ambas as simulações, evidenciando que a nova política de tomada de decisão apresentou maior precisão na seleção das fronteiras, contribuindo para a redução do tempo necessário para a exploração do ambiente. Em todos os critérios de avaliação, a metodologia proposta nesta tese demonstrou superioridade em relação à abordagem de Batinovic et al. [28].

Entre os resultados mais expressivos, destaca-se o tempo total necessário para a execução da missão, que foi 32.83% maior para a metodologia de referência no ambiente *indoor* e 102.52% maior no ambiente *outdoor*. Esse achado é particularmente relevante, pois, apesar de a metodologia de comparação apresentar tempos menores para a detecção de fronteiras, esse fator não impactou de forma significativa o tempo total da exploração autônoma.

Conforme evidenciado ao longo deste trabalho, a detecção de fronteiras só ocorre nos momentos em que o robô precisa tomar decisões sobre sua próxima área de exploração. No entanto, a maior parte do tempo de uma missão autônoma é gasta no deslocamento entre fronteiras para expandir o mapa. Dessa forma, a qualidade da tomada de decisão tem um impacto muito mais significativo na eficiência da exploração autônoma do que o tempo absoluto da detecção de fronteiras.

A independência do tempo necessário para a obtenção das fronteiras foi confirmada por meio das análises de correlação de Pearson e Spearman. Os resultados demonstraram que a metodologia proposta apresentou correlações classificadas como fracas ou muito fracas, enquanto a metodologia de comparação exibiu correlações fortes ou muito fortes. Esse resultado evidencia que o um dos objetivos desta pesquisa – desenvolver uma abordagem que não seja influenciada pela quantidade de fronteiras detectadas – foi plenamente alcançado.

Outro destaque deste trabalho foi a validação prática da metodologia pro-

posta de detecção de fronteiras, utilizando dados registrados em um *rosbag* proveniente de uma missão real de inspeção de tubulações. A robustez da abordagem foi evidenciada pelo fato de que o modelo, treinado com um banco de dados gerado em ambiente simulado, foi capaz de detectar as fronteiras reais de forma eficiente, mesmo em um cenário real com uma configuração de sensores distinta da utilizada no treinamento. Esse resultado demonstra a capacidade de generalização da metodologia, reforçando sua aplicabilidade em diferentes contextos. No entanto, por se tratar de dados históricos, não foi possível atuar sobre o controle do drone durante os experimentos.

Como perspectivas para trabalhos futuros, pretende-se realizar missões práticas com o robô, incorporando sistemas de controle (não abordados neste estudo) para avaliar sua capacidade de exploração do ambiente em condições reais. Além disso, identificou-se a possibilidade de otimizar os *waypoints* gerados pelo planejador de caminhos durante a execução das missões, uma vez que, em muitos casos, as fronteiras associadas aos *waypoints* finais são detectadas antes que o robô atinja seu destino, devido à capacidade dos sensores de identificar obstáculos a distância. Pesquisas futuras poderão expandir essa abordagem, explorando diferentes tipos de terrenos e condições ambientais, ampliando ainda mais sua aplicabilidade.

Uma extensão promissora é estender nossa metodologia a cenários com múltiplos drones operando em conjunto. A ideia é enriquecer a política de decisão hierárquica com informações de posição e disponibilidade de cada plataforma, de modo a alocar fronteiras de exploração de forma dinâmica e balanceada. Paralelamente, a fusão incremental dos mapas gerados por cada VANT aumentaria a qualidade da representação global do ambiente, reduzindo redundâncias e acelerando a descoberta de novas fronteiras. Essa estratégia cooperativa tem potencial para diminuir ainda mais os tempos de exploração e tornar o sistema mais robusto a falhas individuais.

Por fim, outro aprimoramento relevante consiste na incorporação de algoritmos capazes de detectar automaticamente a ausência de áreas efetivas inexploradas, indicando o término da exploração. Uma alternativa promissora é a utilização de algoritmos de planejamento de cobertura (*Coverage Path Planning – CPP*) [127],

que definem um critério de cobertura mínima desejada, encerrando a missão quando esse limiar é atingido.

Em síntese, as contribuições deste trabalho de doutorado representam avanços significativos na exploração autônoma de ambientes desconhecidos, ao integrar redes neurais a uma política de tomada de decisão mais eficiente. A metodologia desenvolvida não apenas reduziu o tempo e a distância percorridos durante as explorações, mas também resultou em uma maior eficiência energética. A continuidade dessa linha de pesquisa tem o potencial de impulsionar o desenvolvimento de sistemas robóticos autônomos mais eficientes, adaptáveis e robustos, ampliando sua aplicabilidade em diferentes cenários de tragédia.

## REFERÊNCIAS

- [1] Faris A Almalki, Ben Othman Soufiene, Saeed H Alsamhi, and Hedi Sakli. A low-cost platform for environmental smart farming monitoring system based on iot and uavs. *Sustainability*, 13(11):5908, 2021.
- [2] Vincenzo Barrile, Silvia Simonetti, Rocco Citroni, Antonino Fotia, and Giuliana Bilotta. Experimenting agriculture 4.0 with sensors: A data fusion approach between remote sensing, uavs and self-driving tractors. *Sensors*, 22(20), 2022.
- [3] Luis F. Luque Vega, Emmanuel Lopez-Neri, Carlos A. Arellano-Muro, Luis E. Gonzalez Jimenez, Jawhar Ghommam, and Rocio Carrasco Navarro. Uav flight instructional design for industry 4.0 based on the framework of educational mechatronics. In *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, pages 2313–2318, 2020.
- [4] Bo Zhang, Guobin Li, Qixin Zheng, Xiaoshan Bai, Yu Ding, and Awais Khan. Path planning for wheeled mobile robot in partially known uneven terrain. *Sensors*, 22(14):5217, 2022.
- [5] Tiedong Liu, Yuxin Sun, Cai Wang, Yangyang Zhang, Zixuan Qiu, Wenfeng Gong, Shuhan Lei, Xinyu Tong, and Xuanyu Duan. Unmanned aerial vehicle and artificial intelligence revolutionizing efficient and precision sustainable forest management. *Journal of Cleaner Production*, 311:127546, 2021.
- [6] Meghna Raj, Shashank Gupta, Vinay Chamola, Anubhav Elhence, Tanya Garg, Mohammed Atiquzzaman, and Dusit Niyato. A survey on the role of internet of things for adopting and promoting agriculture 4.0. *Journal of Network and Computer Applications*, 187:103107, 2021.
- [7] Xin Lyu, Xiaobing Li, Dongliang Dang, Huashun Dou, Kai Wang, and Anru Lou. Unmanned aerial vehicle (uav) remote sensing in grassland ecosystem monitoring: A systematic review. *Remote Sensing*, 14(5), 2022.
- [8] Gabriel GR de Castro, Guido S Berger, Alvaro Cantieri, Marco Teixeira, José Lima, Ana I Pereira, and Milena F Pinto. Adaptive path planning for

- fusing rapidly exploring random trees and deep reinforcement learning in an agriculture dynamic environment uavs. *Agriculture*, 13(2):354, 2023.
- [9] Gabriel GR de Castro, Milena F Pinto, Iago Z Biundini, Aurelio G Melo, Andre LM Marcato, and Diego B Haddad. Dynamic path planning based on neural networks for aerial inspection. *Journal of Control, Automation and Electrical Systems*, 34(1):85–105, 2023.
- [10] Iago Z Biundini, Aurelio G Melo, Fabrício O Coelho, Leonardo M Honório, André LM Marcato, and Milena Faria Pinto. Experimentation and simulation with autonomous coverage path planning for uavs. *Journal of Intelligent & Robotic Systems*, 105(2):46, 2022.
- [11] Petr Stodola, Jan Drozd, Jan Mazal, Jan Hodický, and Dalibor Procházka. Cooperative unmanned aerial system reconnaissance in a complex urban environment and uneven terrain. *Sensors*, 19(17):3754, 2019.
- [12] Jian Zhang and Yang Zhang. A method for uav reconnaissance and surveillance in complex environments. In *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, pages 482–485. IEEE, 2020.
- [13] Hub Ali, Gang Xiong, Muhammad Husnain Haider, Tariku Sinshaw Tamir, Xisong Dong, and Zhen Shen. Feature selection-based decision model for uav path planning on rough terrains. *Expert Systems with Applications*, page 120713, 2023.
- [14] Sabitri Poudel, Muhammad Yeasir Arafat, and Sangman Moh. Bio-inspired optimization-based path planning algorithms in unmanned aerial vehicles: A survey. *Sensors*, 23(6):3051, 2023.
- [15] Sabitri Poudel and Sangman Moh. Hybrid path planning for efficient data collection in uav-aided wsns for emergency applications. *Sensors*, 21(8):2839, 2021.
- [16] Pix4D. Swift action: The importance of drones in disaster response, 2024.
- [17] Pix4D. Interview with a firefighter: How drones saved lives during california’s wildfires, 2024.

- [18] Christian Wankmüller, Maximilian Kunovjanek, and Sebastian Mayrgündter. Drones in emergency response—evidence from cross-border, multi-disciplinary usability tests. *International Journal of Disaster Risk Reduction*, 65:102567, 2021.
- [19] Bogdan Pavkovic, Lazar Berbakov, Sanja Vrane, and Milo Milenkovic. Situation awareness and decision support tools for response phase of emergency management: a short survey. In *2014 25th International Workshop on Database and Expert Systems Applications*, pages 154–159. IEEE, 2014.
- [20] Globo Notícias G1. Vídeo: drone da fab ajuda a resgatar 36 pessoas em 24 horas no rs. <https://g1.globo.com/rs/rio-grande-do-sul/noticia/2024/05/07/video-drone-da-fab-ajuda-a-resgatar-36-pessoas-em-24-horas-no-rs.ghhtml>, May 2024.
- [21] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, pages 146–151. IEEE, 1997.
- [22] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34:189–206, 2013.
- [23] Kai M Wurm, Armin Hornung, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, volume 2, page 3, 2010.
- [24] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot. Informed rrt\*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ international conference on intelligent robots and systems*, pages 2997–3004. IEEE, 2014.

- [25] Fabrício O Coelho, Milena Faria Pinto, Iago Z Biundini, Gabriel GR Castro, Fabio AA Andrade, and Andre LM Marcato. Autonomous uav exploration and mapping in uncharted terrain through boundary-driven strategy. *IEEE Access*, 2024.
- [26] Boyu Zhou, Yichen Zhang, Xinyi Chen, and Shaojie Shen. Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning. *IEEE Robotics and Automation Letters*, 6(2):779–786, 2021.
- [27] Yinghao Zhao, Li Yan, Yu Chen, Hong Xie, and Bo Xu. Faep: Fast autonomous exploration planner for uav equipped with limited fov sensor. *arXiv preprint arXiv:2202.12507*, 2022.
- [28] Ana Batinovic, Tamara Petrovic, Antun Ivanovic, Frano Petric, and Stjepan Bogdan. A multi-resolution frontier-based planner for autonomous 3d exploration. *IEEE Robotics and Automation Letters*, 6(3):4528–4535, 2021.
- [29] Ricardo da Rosa, Marco Aurelio Wehrmeister, Thadeu Brito, José Luís Lima, and Ana Isabel Pinheiro Nunes Pereira. Honeycomb map: a bioinspired topological map for indoor search and rescue unmanned aerial vehicles. *Sensors*, 20(3):907, 2020.
- [30] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40, 1975.
- [31] Steven LaValle. Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*, 1998.
- [32] Cl Connolly. The determination of next best views. In *Proceedings. 1985 IEEE international conference on robotics and automation*, volume 2, pages 432–435. IEEE, 1985.
- [33] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. Receding horizon "next-best-view" planner for 3d exploration. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1462–1468. IEEE, 2016.

- [34] Anna Dai, Sotiris Papatheodorou, Nils Funk, Dimos Tzoumanikas, and Stefan Leutenegger. Fast frontier-based information-driven autonomous exploration with an mav. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 9570–9576. IEEE, 2020.
- [35] Ziyu Cao, Zhihui Du, and Jianhua Yang. Topological map-based autonomous exploration in large-scale scenes for unmanned vehicles. *Drones*, 8(4):124, 2024.
- [36] Jin Wang, Huan Yu, Zhi Zheng, Guodong Lu, Kewen Zhang, Tao Zheng, and Cong Fang. Autonomous robotic exploration with region-biased sampling and consistent decision making. *Complex & Intelligent Systems*, 9(5):6023–6035, 2023.
- [37] Kim-Fung Man, Kit-Sang Tang, and Sam Kwong. Genetic algorithms: concepts and applications [in engineering design]. *IEEE transactions on Industrial Electronics*, 43(5):519–534, 1996.
- [38] Ivan D Changoluisa Caiza, Ana Milas, Marco A Montes Grova, Francisco Javier Perez-Grau, and Tamara Petrovic. Autonomous exploration of unknown 3d environments using a frontier-based collector strategy. *arXiv preprint arXiv:2311.12408*, 2023.
- [39] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007.
- [40] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- [41] Raul Mur-Artal and Juan D. Tardós. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.
- [42] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for

- visual, visual-inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.
- [43] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.
- [44] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.
- [45] Mohammad Amin Sadeghi and David Forsyth. 30hz object detection with dpm v5. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 65–79. Springer, 2014.
- [46] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [47] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [48] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104:154–171, 2013.
- [49] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [50] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

- [51] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [52] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
- [53] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [54] NS Artamonov and PY Yakimov. Towards real-time traffic sign recognition via yolo on a mobile gpu. In *Journal of Physics: Conference Series*, volume 1096, page 012086. IOP Publishing, 2018.
- [55] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.
- [56] Shaoqing Ren, Kaiming He, Ross Girshick, Xiangyu Zhang, and Jian Sun. Object detection networks on convolutional feature maps. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1476–1481, 2016.
- [57] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.
- [58] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

- [59] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [60] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- [61] Chien-Yao Wang, H Liao, Yueh-Hua Wu, Ping-Yang Chen, and IH Yeh. A new backbone that can enhance learning capability of cnn. In *CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE*, pages 390–391, 2020.
- [62] Glenn Jocher. Yolov5 by ultralytics, 2020.
- [63] Yuliang Zhao, Zhongjie Ju, Tianang Sun, Fanghecong Dong, Jian Li, Ruige Yang, Qiang Fu, Chao Lian, and Peng Shan. Tgc-yolov5: An enhanced yolov5 drone detection model based on transformer, gam & ca attention mechanism. *Drones*, 7(7):446, 2023.
- [64] Wojciech Lindenheim-Locher, Adam Świtoński, Tomasz Krzeszowski, Grzegorz Paleta, Piotr Hasiiec, Henryk Josiński, Marcin Paszkuta, Konrad Wojciechowski, and Jakub Rosner. Yolov5 drone detection using multimodal data registered by the vicon system. *Sensors*, 23(14):6396, 2023.
- [65] Bailin Liu and Huan Luo. An improved yolov5 for multi-rotor uav detection. *Electronics*, 11(15):2330, 2022.
- [66] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.
- [67] Chuyi Li, Lulu Li, Yifei Geng, Hongliang Jiang, Meng Cheng, Bo Zhang, Zaidan Ke, Xiaoming Xu, and Xiangxiang Chu. Yolov6 v3. 0: A full-scale reloading. *arXiv preprint arXiv:2301.05586*, 2023.

- [68] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7464–7475, 2023.
- [69] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.
- [70] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023.
- [71] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.
- [72] Yiting Li, Qingsong Fan, Haisong Huang, Zhenggong Han, and Qiang Gu. A modified yolov8 detection network for uav aerial image recognition. *Drones*, 7(5):304, 2023.
- [73] Feng Wang, Hongyuan Wang, Zhiyong Qin, and Jiaying Tang. Uav target detection algorithm based on improved yolov8. *IEEE Access*, 2023.
- [74] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.
- [75] In-So Kweon, M Hebert, Eric Krotkov, and T Kanade. Terrain mapping for a roving planetary explorer. In *IEEE International Conference on Robotics and Automation*, pages 997–1002. IEEE, 1989.
- [76] Rudolph Triebel, Patrick Pfaff, and Wolfram Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *2006 IEEE/RSJ international conference on intelligent robots and systems*, pages 2276–2282. IEEE, 2006.
- [77] Yuval Roth-Tabak and Ramesh Jain. Building an environment model using depth information. *Computer*, 22(6):85–90, 1989.

- [78] Donald Meagher. Geometric modeling using octree encoding. *Computer graphics and image processing*, 19(2):129–147, 1982.
- [79] Pierre Payeur, Patrick Hébert, Denis Laurendeau, and Clément M Gosselin. Probabilistic octree modeling of a 3d dynamic environment. In *Proceedings of International Conference on Robotics and Automation*, volume 2, pages 1289–1296. IEEE, 1997.
- [80] Jonathan Fournier, Benoit Ricard, and Denis Laurendeau. Mapping and exploration of complex environments using persistent 3d model. In *Fourth Canadian Conference on Computer and Robot Vision (CRV'07)*, pages 403–410. IEEE, 2007.
- [81] Manuel Yguel, Olivier Aycard, and Christian Laugier. Update policy of dense maps: Efficient algorithms and sparse representation. In *Field and Service Robotics: Results of the 6th International Conference*, pages 23–33. Springer, 2008.
- [82] Gerhard K Kraetzschmar, Guillem Pages Gassull, and Klaus Uhl. Probabilistic quadtrees for variable-resolution mapping of large environments. *IFAC Proceedings Volumes*, 37(8):675–680, 2004.
- [83] John Amanatides, Andrew Woo, et al. A fast voxel traversal algorithm for ray tracing. In *Eurographics*, volume 87, pages 3–10. Citeseer, 1987.
- [84] Liang Yang, Juntong Qi, Jizhong Xiao, and Xia Yong. A literature review of uav 3d path planning. In *Proceeding of the 11th world congress on intelligent control and automation*, pages 2376–2381. IEEE, 2014.
- [85] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.
- [86] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

- [87] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [88] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. ieee, 1995.
- [89] Marco Dorigo, Gianni Di Caro, and Luca Maria Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1996.
- [90] David E Goldberg. Genetic algorithms in search, optimization, and machine learning. *Addison-Wesley Longman Publishing Co., Inc.*, 412, 1989.
- [91] Junxing Xu, Zengshan Tian, Wei He, and Yaofu Huang. A fast path planning algorithm fusing prm and p-bi-rrt. In *2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan)*, pages 503–508. IEEE, 2020.
- [92] Haitao Gao, Xiaozhu Hou, Jiangpeng Xu, and Banggui Guan. Quad-rotor unmanned aerial vehicle path planning based on the target bias extension and dynamic step size rrt\* algorithm. *World Electric Vehicle Journal*, 15(1):29, 2024.
- [93] Joaquim Ortiz-Haro, Wolfgang Hönig, Valentin N Hartmann, Marc Toussaint, and Ludovic Righetti. idb-rrt: Sampling-based kinodynamic motion planning with motion primitives and trajectory optimization. *arXiv preprint arXiv:2403.10745*, 2024.
- [94] Haiying Liu, Jie Chen, Jianxin Feng, and Haiping Zhao. An improved rrt\* uav formation path planning algorithm based on goal bias and node rejection strategy. *Unmanned Systems*, 11(04):317–326, 2023.
- [95] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.
- [96] James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium*

- Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000.
- [97] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.
- [98] Sebastián Barbas Laina, Simon Boche, Sotiris Papatheodorou, Dimos Tzoumanikas, Simon Schaefer, Hanzhi Chen, and Stefan Leutenegger. Scalable autonomous drone flight in the forest with visual-inertial slam and dense submaps built without lidar. *arXiv preprint arXiv:2403.09596*, 2024.
- [99] Akash Koppam Sreedhara, Deepesh Padala, Shashank Mahesh, Kai Cui, Mengguang Li, and Heinz Koepl. Optimal collaborative transportation for under-capacitated vehicle routing problems using aerial drone swarms. *arXiv preprint arXiv:2310.02726*, 2023.
- [100] Fabrício O Coelho, João P Carvalho, Milena F Pinto, and André L Marcato. Direct-drrt\*: a rrt improvement proposal. In *2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO)*, pages 154–158. IEEE, 2018.
- [101] Maria S Lopes, Manuel F Silva, João Pedro C de Souza, and Pedro Costa. Direct-steered-drrt\*: A 3d rrt-based planner improvement. In *2024 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 100–105. IEEE, 2024.
- [102] David Hsu, Wei S. Lee, and Ari Requicha. Drrt: Dynamic rapidly-exploring random trees for kinodynamic motion planning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 477–483, 2002.
- [103] Intel RealSense D435. <https://www.intelrealsense.com/depth-camera-d435/>. Acesso em: 17 de abril de 2024.

- [104] Ali Safa, Tim Verbelen, Ozan Çatal, Toon Van de Maele, Matthias Hartmann, Bart Dhoedt, and André Bourdoux. Fmcw radar sensing for indoor drones using variational auto-encoders. In *2023 IEEE Radar Conference (RadarConf23)*, pages 1–6. IEEE, 2023.
- [105] Jiexiong Tang, Ludvig Ericson, John Folkesson, and Patric Jensfelt. Gcnv2: Efficient correspondence prediction for real-time slam. *IEEE Robotics and Automation Letters*, 4(4):3505–3512, 2019.
- [106] Dashuai Wang, Wei Li, Xiaoguang Liu, Nan Li, and Chunlong Zhang. Uav environmental perception and autonomous obstacle avoidance: A deep learning and depth camera combined solution. *Computers and Electronics in Agriculture*, 175:105523, 2020.
- [107] Aurelio G Melo, Milena F Pinto, Andre LM Marcato, Leonardo M Honório, and Fabrício O Coelho. Dynamic optimization and heuristics based online coverage path planning in 3d environment for uavs. *Sensors*, 21(4):1108, 2021.
- [108] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 2000.
- [109] Dataset Ninja. Visualization tools for coco 2017 dataset. <https://datasetninja.com/coco-2017>, may 2024. visited on 2024-05-23.
- [110] James B MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.
- [111] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.
- [112] David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, volume 7, pages 1027–1035, 2007.

- [113] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.
- [114] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*. IEEE, 2011.
- [115] Héctor H González-Banos and Jean-Claude Latombe. Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research*, 21(10-11):829–848, 2002.
- [116] Stanford Artificial Intelligence Laboratory et al. Robotic operating system.
- [117] 3D Robotics. IRIS Drone. Online, 2014.
- [118] 3D Robotics. Pixhawk Flight Controller. Online, 2013.
- [119] MAVLink Contributors. MAVLink Protocol. Online, 2022.
- [120] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.
- [121] José Ramiro Martínez de Dios, María Polvillo Díaz, Julio José López Paneque, Víctor Manuel Vega García, and Aníbal Ollero Baturone. Multi-sensor mapping and localization datasets, April 2019.
- [122] Anibal Ollero, Guillermo Heredia, Antonio Franchi, Gianluca Antonelli, Konstantin Kondak, Alberto Sanfeliu, Antidio Viguria, J Ramiro Martinez-de Dios, Francesco Pierri, Juan Cortés, et al. The aeroarms project: Aerial robots with advanced manipulation capabilities for inspection and maintenance. *IEEE Robotics & Automation Magazine*, 25(4):12–23, 2018.

- [123] Velodyne Lidar Inc. *HDL-32E High Definition LiDAR Sensor*. Velodyne Lidar Inc., San Jose, CA, USA, 2013. <https://www.manualslib.com/manual/185894/Velodyne-Lidar-Hdl-32e.html>.
- [124] Karl Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895.
- [125] Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.
- [126] James D Evans. *Straightforward statistics for the behavioral sciences*. Thomson Brooks/Cole Publishing Co, 1996.
- [127] Iago Z Biundini, Milena F Pinto, Aurelio G Melo, Andre LM Marcato, and Leonardo M Honorio. Coverage path planning optimization based on point cloud for structural inspection. *Frontiers in Nature-Inspired Industrial Optimization*, pages 141–156, 2022.