



## UMA ABORDAGEM LEVE PARA DETECÇÃO DE EMBARCAÇÕES UTILIZANDO REDES NEURAIIS SEM PESOS E REGRESSÃO

Adriano Gomes Pereira

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Claudio Miceli de Farias

Rio de Janeiro  
Dezembro de 2025

UMA ABORDAGEM LEVE PARA DETECÇÃO DE EMBARCAÇÕES  
UTILIZANDO REDES NEURAIIS SEM PESOS E REGRESSÃO

Adriano Gomes Pereira

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientador: Claudio Miceli de Farias

Aprovada por: Prof. Leandro Santiago de Araújo  
Prof. Priscila Machado Vieira Lima

RIO DE JANEIRO, RJ – BRASIL  
DEZEMBRO DE 2025

Gomes Pereira, Adriano

Uma abordagem leve para detecção de embarcações utilizando redes neurais sem pesos e regressão/Adriano Gomes Pereira. – Rio de Janeiro: UFRJ/COPPE, 2025.

XIII, 80 p.: il.; 29, 7cm.

Orientador: Claudio Miceli de Farias

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2025.

Referências Bibliográficas: p. 74 – 80.

1. Redes neurais sem pesos. 2. Regressão. 3. Detecção de objetos. 4. Visão computacional. I. Miceli de Farias, Claudio. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

# Agradecimentos

A realização desta dissertação iniciou a partir de uma guinada na minha carreira para um novo caminho: muitos desafios, novos aprendizados e diversas conquistas que não seriam possíveis sem o apoio, a dedicação e a inspiração de pessoas especiais que caminharam ao meu lado ao longo dessa jornada.

Em primeiro lugar, agradeço profundamente a Deus. Foram inúmeras orações, conversas e momentos de reflexão em busca de orientação e consolo diante dos desafios que surgiram antes mesmo do meu ingresso no programa. Agradeço também à Maria, minha mãezinha, pela sua constante intercessão, por me acolher em suas preces e por me lembrar, nos momentos de incerteza, da importância da fé, da paciência e da confiança nos planos de Deus.

Em seguida, agradeço profundamente aos meus pais, Carlos e Fátima Pereira, que sempre foram o alicerce da minha vida. Com seu exemplo de trabalho, honestidade e perseverança, mostraram-me desde cedo a importância do esforço contínuo, da disciplina e da educação como ferramentas de transformação. Foram eles que me deram não apenas as condições materiais, mas, sobretudo, o amor incondicional, o incentivo moral e o suporte emocional para que eu pudesse chegar até aqui. Este trabalho é, em grande medida, fruto do carinho, da dedicação e do cuidado que recebi em cada etapa do meu crescimento.

À minha esposa Flávia Pereira, expresso uma gratidão imensurável. Nenhum agradecimento seria suficiente para traduzir a dimensão da nossa parceria e de tudo o que construímos juntos ao longo desses quase 17 anos. Durante esta jornada, vem sendo meu porto seguro, oferecendo apoio nos momentos mais difíceis e celebrando comigo cada pequena conquista. Sua paciência, compreensão e incentivo constantes foram essenciais para que eu mantivesse o equilíbrio e seguisse em frente, mesmo frente aos desafios. É ao seu lado que compartilho não apenas esta conquista acadêmica, mas a vida em sua totalidade, com a certeza de que nossa caminhada conjunta se fortalece e se renova a cada dia. Caminhada esta, tão leve, plena e cheia de amor, que agora se expande com a nossa filha a caminho, trazendo ainda mais sentido e alegria aos nossos dias. Obrigado por estar aqui.

Agradeço também ao meu irmão, Rafael Pereira. Companheiro de tantas histórias, sempre me trouxe incentivo, apoio e uma amizade verdadeira. Sua presença

constante, sua torcida e seus conselhos foram fundamentais para que eu mantivesse a confiança em meus próprios passos. Sou profundamente grato por tê-lo como irmão e me sinto no dever de estar sempre à altura dessa relação, buscando ser um exemplo positivo e uma influência que o inspire a se tornar, a cada dia, uma pessoa ainda melhor. Nossa convivência desde a infância moldou quem eu sou e me ensinou o valor da fraternidade, da lealdade e do amor entre irmãos.

Ao meu orientador, Claudio Miceli, deixo registrado meu sincero reconhecimento e profundo agradecimento. Sua orientação firme, generosa e criteriosa guiou não apenas a execução desta dissertação, mas também de outros artigos e o meu amadurecimento como pesquisador. No meio desse caminho acadêmico, tive a felicidade de ser acolhido por ele, que prontamente me recebeu como orientando e acreditou no meu trabalho. Desde então, demonstra constante disponibilidade para compartilhar conhecimento, oferecendo direcionamento preciso, confiança e incentivo em cada etapa do processo. Sua orientação foi essencial para que eu superasse minhas limitações, que não eram poucas, encontrasse novos caminhos e desenvolvesse um trabalho pelo qual serei sempre grato.

Agradeço também ao Professor Leandro Santiago e à Professora Priscila Lima por gentilmente aceitarem o convite para compor a banca examinadora. Expresso minha gratidão pelos ensinamentos e pelas discussões sobre redes neurais sem pesos, que despertaram em mim o interesse por essa área e contribuíram diretamente para o direcionamento e amadurecimento desta pesquisa.

Aos meus amigos, deixo registrado meu mais sincero agradecimento. Cada um, à sua maneira, tornou esta jornada mais leve e significativa. Agradeço especialmente à Matheus Rocha e Jader Esteves pela presença constante, pelas conversas que renovam meu ânimo e pelas risadas que trazem equilíbrio aos dias complicados. A amizade de todos vocês é abrigo nos momentos de incerteza e celebração nas conquistas. Sou grato por ter ao meu lado pessoas que torcem verdadeiramente por mim, que compartilham sonhos, planos e histórias. Nossa convivência me ensinou que a amizade é um dos maiores sustentáculos da vida, um laço que se fortalece no tempo e resiste à distância. Que este trabalho, de alguma forma, também carregue um pouco da energia, da alegria e do carinho que cada um de vocês me proporcionou ao longo desse caminho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## UMA ABORDAGEM LEVE PARA DETECÇÃO DE EMBARCAÇÕES UTILIZANDO REDES NEURAI SEM PESOS E REGRESSÃO

Adriano Gomes Pereira

Dezembro/2025

Orientador: Claudio Miceli de Farias

Programa: Engenharia de Sistemas e Computação

A detecção automática de embarcações em ambientes marítimos representa um desafio significativo na visão computacional, devido à variabilidade de condições ambientais, reflexos da superfície da água e diversidade de perfis e proporções dos navios. Este trabalho propõe o desenvolvimento de um sistema leve e eficiente para a detecção de embarcações em imagens óticas, denominado *SOD-Edge (Single Object Detection for Edge Computing)*, com foco na viabilidade em sistemas embarcados de baixo consumo energético. A metodologia combina a extração de características a partir de redes neurais convolucionais eficientes com o uso de Redes Neurais Sem Pesos (WiSARD) para classificação e diferentes estratégias de regressão para a estimativa dos parâmetros das *bounding boxes*, incluindo *Support Vector Regression (SVR)*, *k-Nearest Neighbors (kNN)* e *Regression Wisard (ReWiSARD)*.

Foram conduzidos experimentos em dois ambientes computacionais distintos: um ambiente em nuvem (Google Colab) e um dispositivo embarcado (Raspberry Pi 4), de modo a avaliar não apenas métricas de desempenho como acurácia, F1-score, MAE e IoU, mas também eficiência em termos de tempo de execução e consumo energético. Os resultados demonstraram que a combinação entre CNNs eficientes, WiSARD e regressão leve é capaz de fornecer desempenho competitivo em relação a modelos baseados exclusivamente em arquiteturas profundas, apresentando simultaneamente vantagens em simplicidade estrutural, eficiência temporal e consumo energético.

Este estudo reforça a relevância da exploração de alternativas às redes profundas convencionais em cenários de restrição computacional, além de evidenciar que a escolha do método de regressão pode ser orientada pelos requisitos da aplicação. Por fim, o trabalho abre perspectivas para futuras extensões voltadas à detecção

de múltiplos objetos por imagem e à análise de sequências de vídeo em ambientes marítimos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

## A LIGHTWEIGHT APPROACH FOR SHIP DETECTION USING WEIGHTLESS NEURAL NETWORKS AND REGRESSION

Adriano Gomes Pereira

December/2025

Advisor: Claudio Miceli de Farias

Department: Systems Engineering and Computer Science

Automatic ship detection in maritime environments remains a challenging task in computer vision, mainly due to the variability of environmental conditions, water surface reflections, and the diversity of vessel profiles and proportions. This dissertation proposes the development of a lightweight and efficient system for ship detection in optical images, named SOD-Edge (Single Object Detection for Edge Computing), targeting deployment in embedded systems with limited energy resources. The methodology combines feature extraction from efficient convolutional neural networks with Weightless Neural Networks (WiSARD) for classification and multiple regression strategies for bounding box estimation, including Support Vector Regression (SVR), k-Nearest Neighbors (kNN), and a regression-based variant of WiSARD (ReWiSARD).

Experiments were conducted in two different computational environments: a cloud-based setup (Google Colab) and an embedded device (Raspberry Pi 4). The evaluation considered not only performance metrics such as accuracy, F1-score, MAE, and IoU, but also efficiency regarding runtime and energy consumption. Results show that the combination of efficient CNNs, WiSARD, and lightweight regression methods provides competitive performance compared to purely deep learning-based approaches, while offering advantages in terms of structural simplicity, runtime efficiency, and reduced energy consumption.

This study highlights the relevance of exploring alternatives to conventional deep neural networks in constrained computational scenarios and demonstrates that the choice of the regression method can be guided by application-specific requirements. Finally, it opens perspectives for future extensions involving multi-object detection and video analysis in maritime environments.

# Sumário

<b>Lista de Figuras</b>	<b>xi</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Problema de Pesquisa . . . . .	9
1.2 Hipóteses do Trabalho . . . . .	9
<b>2 Conceitos Básicos</b>	<b>14</b>
2.1 Visão Computacional em Ambientes Marítimos . . . . .	14
2.2 Redes Neurais em Visão Computacional . . . . .	16
2.2.1 Extração de Características e <i>Transfer Learning</i> . . . . .	18
2.3 Redes Neurais Sem Pesos (WNNs) . . . . .	20
2.3.1 WiSARD com <i>Bleaching</i> . . . . .	21
2.3.2 Regression WiSARD (ReWiSARD) . . . . .	22
2.4 Support Vector Regression (SVR) . . . . .	23
2.5 k-Nearest Neighbors (kNN) . . . . .	25
2.6 Resumo do Capítulo . . . . .	26
<b>3 Trabalhos Relacionados</b>	<b>28</b>
3.1 Introdução . . . . .	28
3.2 WiSARD e Redes Neurais Sem Peso . . . . .	28
3.3 Support Vector Regression em Visão Computacional . . . . .	30
3.4 Detecção de Embarcações com Redes Neurais Profundas . . . . .	31
<b>4 Estratégia de Detecção de Objeto Único em Borda</b>	<b>33</b>
4.1 Visão Geral da Proposta . . . . .	33
4.2 Formulação do Problema . . . . .	36
4.3 Estruturas de Dados . . . . .	37
4.3.1 Imagem de entrada e pré-processamento . . . . .	37
4.3.2 Extração de características com CNN leve . . . . .	37
4.3.3 Normalização e codificação dos atributos . . . . .	39

4.3.4	Estrutura dos rótulos . . . . .	39
4.4	Pipeline Proposto . . . . .	40
4.4.1	Pipeline de Preparação . . . . .	42
4.4.2	Pipeline de Detecção . . . . .	42
4.4.3	Entrada e saída do modelo . . . . .	44
<b>5</b>	<b>Experimentos e Resultados</b>	<b>46</b>
5.1	Implementação . . . . .	46
5.1.1	Recursos utilizados . . . . .	46
5.1.2	Como foi implementado . . . . .	47
5.1.3	Limitações . . . . .	49
5.2	Cenários de Experimentação e Dados . . . . .	49
5.2.1	Ambientes de execução . . . . .	49
5.2.2	Base de dados . . . . .	50
5.2.3	Justificativa da Escolha . . . . .	51
5.3	Métricas . . . . .	51
5.3.1	Métricas de Classificação . . . . .	51
5.3.2	Métricas de Regressão . . . . .	52
5.3.3	Métricas de Eficiência Temporal . . . . .	53
5.3.4	Métricas de Eficiência Energética . . . . .	53
5.3.5	Resumo das Métricas . . . . .	54
5.4	Experimentos . . . . .	54
5.4.1	Experimento 1: Classificação . . . . .	55
5.4.2	Experimento 2: Regressão das caixas delimitadoras . . . . .	55
5.4.3	Experimento 3: Eficiência Temporal . . . . .	55
5.4.4	Experimento 4: Eficiência Energética . . . . .	55
5.5	Avaliação de Resultados . . . . .	56
5.5.1	1º Experimento – Classificação . . . . .	56
5.5.2	2º Experimento – Regressão . . . . .	57
5.5.3	3º Experimento – Eficiência Temporal . . . . .	59
5.5.4	4º Experimento – Eficiência Energética . . . . .	61
5.5.5	Discussão Geral . . . . .	64
<b>6</b>	<b>Conclusão</b>	<b>67</b>
6.1	Visão geral do Trabalho . . . . .	67
6.2	Limitações do Trabalho . . . . .	69
6.3	Trabalhos Futuros . . . . .	70
6.4	Considerações Finais . . . . .	71
	<b>Referências Bibliográficas</b>	<b>74</b>

# Lista de Figuras

1.1	Exemplo de um ambiente marítimo com embarcações em diferentes condições. . . . .	3
2.1	Diferentes perspectivas de um mesmo tipo de embarcação: vistas de través, frontal e traseira. . . . .	15
2.2	Arquitetura da rede VGG-16, ilustrando a progressão de camadas convolucionais, operações de <i>pooling</i> e camadas totalmente conectadas. 17	
2.3	Exemplo de treinamento de uma rede WiSARD: a imagem é dividida em 4 tuplas de tamanho 4 e cada uma representa uma RAM. Cada vez que o respectivo endereço da tupla é acessado, seu valor na tabela vai de zero para um. Fonte: SANTIAGO <i>et al.</i> (2020) . . . . .	21
2.4	Exemplo de regressão por vetores de suporte (SVR). Ilustração de uma função de regressão SVR representada por $w^T x$ . O tubo $\epsilon$ -insensitive em torno da função é indicado em cinza. $\xi_i = w^T x_i$ corresponde ao valor previsto para a amostra $x_i$ , enquanto $y_i$ representa o valor real. Os vetores de suporte estão destacados com borda vermelha. Fonte: ROSENBAUM <i>et al.</i> (2013) . . . . .	25
4.1	Pipeline proposto para detecção de embarcações. A imagem de entrada passa por pré-processamento e extração de características com CNN; as <i>features</i> seguem dois fluxos: (i) binarização e classificação pela WiSARD e (ii) codificação e regressão das <i>bounding boxes</i> . Ambos convergem para uma saída única contendo a classe e a localização da embarcação. . . . .	36
4.2	Na codificação de termômetro, os bits são ativados progressivamente do menos significativo para o mais significativo, de forma análoga à subida do mercúrio em um termômetro. Na figura, o valor "0.75" é codificado em um termômetro 0 à 1 de resolução 10. . . . .	39
4.3	Exemplo de vetor One-Hot para sete classes de embarcações. Apenas a posição correspondente à classe é ativada, enquanto todas as demais permanecem inativas. Na figura, a classe "Veleiro" é selecionada. . . .	39

5.1	Matriz de confusão dos resultados de classificação com a rede WiSARD, onde as linhas representam as classes reais e as colunas indicam as predições. As células diagonais evidenciam as instâncias corretamente classificadas, enquanto os valores fora da diagonal revelam confusões entre classes visualmente semelhantes, navios militares e patrulha. . . . .	57
5.2	Resultados qualitativos da abordagem proposta utilizando o modelo SVR como regressor. As caixas em verde representam as anotações do <i>ground truth</i> (GT), enquanto as caixas em vermelho indicam as predições obtidas pelo modelo. . . . .	64

# Lista de Tabelas

5.1	Distribuição de classes no conjunto de dados . . . . .	50
5.2	Resumo das métricas utilizadas nos experimentos e seus objetivos. . .	54
5.3	Resultados médios de classificação com WiSARD e YOLO . . . . .	56
5.4	Resultados da regressão dos parâmetros da <i>bounding box</i> utilizando SVR, kNN, ReWiSARD e YOLOv8. . . . .	58
5.5	Custo temporal (em segundos) do pipeline proposto utilizando WiSARD combinada com SVR, kNN e ReWiSARD, em comparação ao YOLOv8. . . . .	60
5.6	Custo temporal médio (em segundos), com desvio padrão, da solução proposta no ambiente embarcado (Raspberry Pi 4). . . . .	61
5.7	Consumo energético (em mWh) das diferentes etapas do pipeline proposto, comparando a WiSARD (classificação) e os algoritmos de regressão SVR, kNN e ReWiSARD com o YOLOv8. . . . .	62
5.8	Consumo energético (em mWh) da solução proposta no ambiente embarcado (Raspberry Pi 4). . . . .	64

# Capítulo 1

## Introdução

A detecção de embarcações em ambientes marítimos é um desafio em diversas áreas em aplicações tecnológica e econômica. Sistemas automatizados de monitoramento e vigilância marítima desempenham um papel estratégico e fundamental em atividades como segurança, patrulhamento de fronteiras, combate à pesca ilegal, controle de tráfego marítimo, gerenciamento ambiental, além do apoio a operações de busca e salvamento ZHAO *et al.* (2024). Com o aumento do volume de embarcações circulando em áreas costeiras e oceânicas, impulsionado pela globalização do comércio e pelo crescimento das atividades marítimas civis e militares, a demanda por soluções computacionais eficazes e autônomas se intensifica consideravelmente.

Os sistemas de radar costeiro constituem uma das ferramentas mais tradicionais, empregados em operações de vigilância e controle do tráfego marítimo, sendo capazes de detectar embarcações a longas distâncias de forma contínua, embora apresentem menor capacidade para distinguir alvos de pequeno porte. O sistema de Identificação Automática (AIS), por sua vez, baseia-se em transponders instalados a bordo das embarcações, transmitindo periodicamente dados como posição, velocidade, rumo e identidade, o que possibilita um monitoramento preciso das embarcações registradas, mas não contempla aquelas que operam de forma irregular ou não cooperativa. Além desses meios, recursos visuais têm ganhado relevância crescente: câmeras instaladas em pontos estratégicos da costa permitem acompanhamento local em alta resolução, ainda que restrito ao alcance geográfico da instalação; imagens de satélite oferecem ampla cobertura espacial, favorecendo análises em grande escala, mas com limitações ligadas à periodicidade de captura e às condições climáticas; e veículos aéreos não tripulados (VANTs) proporcionam flexibilidade para alcançar áreas de difícil acesso ou patrulhamento convencional, embora dependam de autonomia energética limitada e de infraestrutura para operação segura. Essa diversidade de soluções evidencia o caráter multidisciplinar do monitoramento aquático e abre espaço para estratégias híbridas, capazes de combinar diferentes fontes de informação em sistemas integrados de vigilância e análise.

Apesar da diversidade de tecnologias empregadas no monitoramento aquático, os métodos baseados em visão computacional assumem papel central por possibilitarem a identificação direta de embarcações e pessoas a partir de imagens. Contudo, sua aplicação nesse contexto apresenta um conjunto particular de desafios, uma vez que as condições ambientais em ambientes aquáticos diferem substancialmente daquelas observadas em cenários urbanos ou industriais.

O monitoramento de espaços aquáticos, diferentemente de cenários urbanos ou industriais, envolve condições desafiadoras para sistemas de visão computacional. Refração da luz, reflexos na superfície da água, nebulosidade, variações bruscas de iluminação, presença de ondas e a ausência de referências fixas tornam o processo de detecção visual mais complexo e sujeito a falhas MOOSBAUER *et al.* (2019). Além disso, as embarcações apresentam ampla variabilidade em suas formas, tamanhos, cores e velocidades, o que dificulta a definição de padrões rígidos de identificação. Detectar pequenos barcos em grandes áreas abertas, por exemplo, exige precisão e robustez de algoritmos para evitar tanto falsos positivos quanto omissões.

Os métodos tradicionais de visão computacional desempenharam papel importante em aplicações de vigilância marítima, especialmente em cenários anteriores à popularização das redes neurais profundas. Essas abordagens baseavam-se em técnicas de processamento de imagem e extração manual de características, como operadores de borda, como os filtros de Sobel e de Canny ZHU *et al.* (2016); ALVEAR-SANDOVAL *et al.* (2017) para realce de contornos de embarcações, transformadas geométricas para detecção de linhas e formas, e descritores de textura ou gradiente, como HOG (*Histogram of Oriented Gradients*) e SIFT (*Scale-Invariant Feature Transform*), aplicados na identificação de objetos em diferentes escalas e orientações DALAL e TRIGGS (2005); LOWE (2004). Em ambientes aquáticos, também foram explorados métodos de segmentação baseados em limiarização, diferenças de movimento (*background subtraction*) e *clustering* de regiões, buscando isolar embarcações do fundo dinâmico formado pelo mar HU *et al.* (2004). Embora essas técnicas tenham fornecido resultados relevantes em determinados contextos, sua dependência de condições ambientais controladas e a limitada capacidade de generalização frente à variabilidade de cenários marítimos acabaram impulsionando a adoção de abordagens mais robustas, baseadas em aprendizado de máquina e, posteriormente, em redes neurais profundas WANG *et al.* (2023).

Nos últimos anos, o avanço da inteligência artificial, especialmente das técnicas baseadas em aprendizado profundo LECUN *et al.* (2015), permitiu grandes progressos na detecção de objetos em imagens e vídeos. Modelos como as Redes Neurais Convolucionais (CNNs) KRIZHEVSKY *et al.* (2012) e suas variações evoluíram significativamente, sendo empregados com sucesso em tarefas de classificação e detecção em contextos terrestres, aéreos e marítimos. Entre os principais algoritmos

de detecção de objetos, destacam-se arquiteturas como Faster R-CNN, *Single Shot Multibox Detector* (SSD) REN *et al.* (2015); LIU *et al.* (2016) e a família YOLO (*You Only Look Once*) REDMON *et al.* (2016), que alcançam resultados notáveis em *benchmarks* visuais.

Particularmente, o algoritmo YOLOv8, a versão mais recente da série, representa o estado da arte em termos de equilíbrio entre acurácia e velocidade de inferência. Ele é amplamente utilizado em aplicações práticas de monitoramento em tempo real, oferecendo suporte a múltiplas resoluções de entrada e otimizações para diferentes plataformas. Contudo, apesar de sua eficiência relativa, modelos como o YOLOv8 ainda requerem uma infraestrutura computacional considerável para alcançar desempenho satisfatório. Essa dependência implica não apenas a necessidade de *hardware* especializado, mas também custos elevados de aquisição, consumo energético e manutenção, o que torna tais soluções pouco realistas para cenários com recursos limitados. Em dispositivos de borda, drones, robôs autônomos ou embarcações não tripuladas, onde o espaço físico, a disponibilidade energética e a capacidade de processamento são severamente restritos, a adoção de arquiteturas pesadas de aprendizado profundo se torna inviável. Nesses contextos, a necessidade de alternativas mais leves e energeticamente eficientes torna-se evidente, de modo a viabilizar o monitoramento autônomo sem depender de infraestrutura computacional robusta e dispendiosa.



Figura 1.1: Exemplo de um ambiente marítimo com embarcações em diferentes condições.

Apesar dos avanços substanciais proporcionados pelas redes neurais profundas, suas aplicações em contextos com restrições de hardware ainda enfrentam limitações importantes. A maior parte das arquiteturas de detecção de objetos amplamente

utilizadas — como YOLOv8, Faster R-CNN e SSD — foi desenvolvida e otimizada tendo como premissa a disponibilidade de ambientes computacionais robustos, muitas vezes com suporte a unidades de processamento gráfico (GPUs), grandes quantidades de memória e acesso contínuo à energia elétrica e conectividade com servidores em nuvem.

No entanto, essas condições nem sempre estão presentes em aplicações reais, especialmente naquelas inseridas no escopo da Internet das Coisas (*Internet of Things* – IoT), da computação na borda (*Edge Computing*) e de sistemas autônomos embarcados MURSHED *et al.* (2019). Dispositivos como drones, robôs de superfície (USVs), boias inteligentes ou plataformas de monitoramento ambiental operam, frequentemente, em ambientes hostis, isolados ou móveis, onde o acesso à infraestrutura tradicional de computação é inexistente ou altamente limitado CHEN *et al.* (2020). Nesses cenários, é necessário que os algoritmos de visão computacional sejam não apenas precisos, mas também altamente eficientes em termos de processamento, consumo energético e velocidade de resposta SORET *et al.* (2021).

Modelos como YOLOv8, embora mais leves do que seus predecessores, ainda requerem dispositivos com unidades de processamento vetorial, suporte a CUDA ou outras bibliotecas de aceleração, e consumo energético que pode ser proibitivo para longas operações em campo. A tentativa de adaptar esses modelos para plataformas como o Raspberry Pi, por meio de quantização ou uso de versões reduzidas (como YOLOv5n ou YOLOv8n), frequentemente resulta em perdas significativas de desempenho, ou em uma redução apenas parcial das demandas computacionais.

Além disso, o processo de treinamento desses modelos costuma ser altamente dependente de grandes conjuntos de dados rotulados, o que representa uma barreira adicional em áreas onde a coleta e anotação de dados é difícil ou dispendiosa. Para problemas específicos como a detecção de embarcações — em que o volume de imagens disponíveis pode ser limitado, e as condições visuais variam amplamente — a dependência de dados massivos se torna um obstáculo à replicabilidade e escalabilidade das soluções baseadas em aprendizado profundo.

Outro aspecto relevante diz respeito à opacidade dos modelos de redes profundas. Apesar de seu alto desempenho, essas arquiteturas funcionam, na prática, como “caixas-pretas”, dificultando a interpretação dos resultados e a análise de erros. Em aplicações sensíveis, como segurança marítima ou vigilância ambiental, a capacidade de entender e justificar decisões do sistema é um fator crucial para a aceitação e confiabilidade da tecnologia.

Com a expansão da IoT e da chamada Edge AI — inteligência artificial operando localmente em dispositivos com capacidade computacional limitada — torna-se cada vez mais importante desenvolver modelos que conciliem eficiência energética, baixo custo computacional e bom desempenho em tarefas complexas SINGH *et al.* (2023);

SURIANARAYANAN *et al.* (2023). Nesse contexto, a detecção de embarcações em tempo real em sistemas embarcados representa uma demanda crítica, com implicações diretas para a autonomia de veículos marítimos, a escalabilidade de soluções de vigilância e a viabilidade de aplicações de longo prazo em regiões remotas ou com conectividade limitada WANG *et al.* (2025a).

A capacidade de executar algoritmos de visão computacional em sistemas de baixo custo abre possibilidades significativas para monitoramento em tempo real, mesmo em cenários onde não há acesso constante à internet ou a servidores em nuvem. Contudo, os limites impostos pelo *hardware* — restrições de memória, processamento e energia — exigem o uso de algoritmos leves, de rápida inferência e baixa dependência de recursos externos LIU *et al.* (2025).

Diante desse panorama, torna-se evidente a necessidade de explorar abordagens alternativas aos modelos tradicionais de aprendizado profundo, especialmente aquelas capazes de operar eficientemente em ambientes computacionalmente restritos. A busca por soluções compatíveis com os requisitos da IoT e que mantenham a capacidade de detecção eficaz em contextos desafiadores motiva o presente trabalho, cuja proposta se insere nessa interseção entre eficiência, aplicabilidade e inovação tecnológica.

A evolução das técnicas de visão computacional para cenários de recursos restritos tem sido impulsionada pelo avanço de arquiteturas e estratégias voltadas ao campo do TinyML. Diferentemente do *deep learning* convencional, que depende de modelos de grande porte executados em infraestrutura baseada em GPUs, o TinyML busca desenvolver soluções capazes de operar em microcontroladores e sistemas embarcados de baixo consumo energético. Nesse contexto, surgiram variações compactas das Redes Neurais Convolucionais (CNNs), projetadas para extrair automaticamente características visuais relevantes com menor custo computacional. Exemplos incluem arquiteturas como a MobileNet HOWARD *et al.* (2017) HOWARD *et al.* (2017) e a SqueezeNet IANDOLA *et al.* (2016), que exploram convoluções fatorizadas, camadas de profundidade separável e estratégias de compressão de parâmetros para reduzir significativamente a complexidade do modelo, mantendo desempenho competitivo.

Na tarefa específica de detecção de objetos, diferentes adaptações foram propostas para viabilizar a execução em dispositivos de borda. Modelos de duas etapas, como o Faster R-CNN, apresentaram resultados precisos, mas com elevado custo computacional, o que os torna pouco adequados a esse cenário. Em contrapartida, arquiteturas de detecção em uma única etapa, como as versões compactas da família YOLO, mostraram-se mais promissoras, pois simplificam o processo de detecção ao realizá-lo diretamente sobre a imagem completa. Avanços recentes, como o YOLOv5-Nano e o YOLOv8-N, exemplificam esse movimento de adaptação, oferecendo um

equilíbrio entre acurácia e eficiência, ainda que, em muitos casos, dependam de aceleração por GPU para alcançar desempenho satisfatório.

Esse contexto evidencia a necessidade de modelos ainda mais leves e energeticamente eficientes, capazes de explorar plenamente o potencial do TinyML. A busca por técnicas que conciliem baixo consumo de recursos, robustez e aplicabilidade em tempo real em dispositivos de borda é hoje um dos principais vetores de pesquisa em visão computacional embarcada.

Apesar de suas qualidades, mesmo os modelos mais otimizados da família YOLO ainda demandam recursos computacionais significativos. A complexidade das operações matriciais e das convoluções profundas dificulta sua execução em ambientes restritos, como microcontroladores ou dispositivos embarcados sem suporte a aceleração por *hardware*. Além disso, o processo de treinamento desses modelos depende de grandes volumes de dados anotados e de infraestrutura computacional robusta, o que representa uma barreira para cenários que exigem implementação rápida ou dispõem de recursos limitados. Torna-se então necessária a adoção de abordagens mais leves e energeticamente eficientes, capazes de viabilizar a visão computacional em dispositivos de baixo custo e baixo consumo. É justamente nesse espaço que se insere o TinyML ABADADE *et al.* (2023), cuja proposta é desenvolver e implantar modelos de aprendizado de máquina adaptados a ambientes com forte restrição de memória, processamento e energia, permitindo a aplicação prática de técnicas de inteligência artificial em sistemas embarcados.

Diante desses desafios, há uma crescente necessidade por modelos que ofereçam transparência, simplicidade e viabilidade computacional, sem comprometer significativamente o desempenho em tarefas de detecção. Essa necessidade motiva a investigação de alternativas como as Redes Neurais Sem Pesos (*Weightless Neural Networks* - WNNs) ALEKSANDER *et al.* (1984), que se destacam por sua estrutura compacta, baixa exigência de dados para treinamento, e maior interpretabilidade dos processos internos — características particularmente desejáveis em sistemas embarcados de visão computacional.

Modelos de WNN, ao invés de aprender pesos ajustáveis por meio de *backpropagation*, utilizam estruturas simples e interpretáveis para codificar informações. A eliminação de operações aritméticas pesadas e o uso de estruturas como tabelas de busca ou memórias RAM binárias conferem às WNNs uma complexidade computacional extremamente baixa, o que as torna candidatas ideais para aplicações embarcadas, em tempo real e com restrições energéticas.

Uma das arquiteturas mais representativas desse paradigma é a WiSARD (Wilkie, Stonham and Aleksander's Recognition Device), introduzida inicialmente para tarefas de reconhecimento de padrões binários. Sua operação, baseada em unidades de memória endereçável e codificações binárias simples, permite implementações

extremamente leves e adaptáveis. Ao mesmo tempo, a arquitetura preserva a capacidade de aprendizado incremental e rápido, com baixa dependência de dados massivos ou processos de treinamento extensos.

A literatura científica, embora mais limitada em comparação com os estudos sobre CNNs, já apresenta aplicações bem-sucedidas da WiSARD em tarefas como classificação de texto, reconhecimento de dígitos manuscritos e até mesmo problemas biomédicos. Contudo, seu uso em visão computacional, especialmente em tarefas de detecção de objetos PEREIRA *et al.* (2025), ainda é um campo emergente, com grande potencial para ser explorado, sobretudo em conjunto com técnicas de pré-processamento e codificação adequadas às características visuais contínuas, como a codificação termométrica.

Dessa forma, observa-se uma lacuna significativa entre os modelos dominantes baseados em aprendizado profundo e alternativas mais leves que possam ser aplicadas em sistemas embarcados de forma eficiente. Essa lacuna motiva a exploração de arquiteturas como a WiSARD, que, apesar de sua simplicidade, apresenta vantagens promissoras no contexto da IoT e da Inteligência Artificial na borda (*Edge AI* SINGH e GILL (2023)).

Complementarmente, para a tarefa de regressão das coordenadas de localização das embarcações, este trabalho investigou diferentes métodos que atendessem aos requisitos de baixo custo computacional, rápida inferência e adequação a dispositivos embarcados. Nesse contexto, a abordagem inicialmente adotada foi a *Regression* WiSARD, uma extensão das Redes Neurais Sem Pesos para problemas de regressão, fundamentada em aprendizado baseado em memória e codificação binária. Essa escolha foi motivada por resultados previamente obtidos pelo autor em trabalhos publicados, bem como pela compatibilidade conceitual da *Regression* WiSARD com a etapa de classificação baseada na WiSARD, permitindo a construção de um sistema coeso e inteiramente orientado a modelos leves.

De forma complementar, e com o objetivo de ampliar a análise comparativa, este trabalho também avaliou os métodos *k-Nearest Neighbors* (kNN) e *Support Vector Regression* (SVR). O algoritmo kNN foi considerado por sua simplicidade conceitual e custo de treinamento baixo, enquanto o SVR, uma extensão das Máquinas de Vetores de Suporte (*Support Vector Machine* – SVM) para predição de variáveis contínuas, foi incluído por sua robustez teórica e ampla utilização em problemas de regressão na literatura. A análise conjunta desses três métodos — *Regression* WiSARD, kNN e SVR — permite investigar diferentes paradigmas de regressão sob a ótica da eficiência computacional, da acurácia e da viabilidade para aplicações de detecção de objetos em ambientes embarcados.

Considerando os desafios impostos pelos métodos tradicionais de detecção de objetos em ambientes com restrições computacionais, esta dissertação propõe uma

abordagem alternativa para a detecção de embarcações em imagens marítimas, baseada na combinação de WNNs com métodos leves de regressão. A proposta central consiste em empregar a arquitetura WiSARD para a tarefa de classificação das embarcações e acoplar a essa etapa um módulo de regressão responsável pela estimação das coordenadas das caixas delimitadoras.

A arquitetura WiSARD, concebida inicialmente para o reconhecimento de padrões binários, fundamenta-se no uso de memórias RAM endereçáveis, que registram padrões de entrada de forma não paramétrica. Ao contrário das redes neurais tradicionais, que dependem da aprendizagem de pesos por meio de iterações de retropropagação de erro, o modelo WiSARD realiza aprendizado associativo direto, o que reduz drasticamente a necessidade de recursos computacionais e acelera significativamente os tempos de treinamento e inferência. Essa característica torna a arquitetura especialmente atraente para aplicações embarcadas, onde a agilidade no processamento e a economia de energia são requisitos centrais.

Para adaptar a WiSARD ao domínio da visão computacional, especialmente à análise de imagens contínuas, o presente trabalho incorpora o uso de codificação termométrica (*thermometer encoding*) — uma técnica que converte atributos contínuos em vetores binários, preservando a ordenação natural dos dados e otimizando sua representação para as memórias endereçáveis da rede. Essa etapa é essencial para o mapeamento adequado dos valores de pixels ou atributos extraídos das imagens para o formato binário exigido pela WiSARD, sem perda significativa de informação.

Para o módulo de regressão, inicialmente é instanciado utilizando a *Regression* WiSARD, explorando a compatibilidade conceitual entre os modelos sem pesos e seu potencial para aplicações embarcadas. De forma complementar, os métodos kNN e SVR também são investigados, permitindo uma análise comparativa entre diferentes paradigmas de regressão sob os critérios de acurácia, custo computacional e viabilidade para execução em tempo real. Essa solução híbrida busca validar o uso de modelos computacionalmente leves, interpretáveis e eficientes, capazes de operar em dispositivos embarcados, como o Raspberry Pi, oferecendo uma alternativa prática ao uso de arquiteturas profundas tradicionais.

Além do foco no desempenho algorítmico, esta dissertação enfatiza a validação prática da proposta em um ambiente embarcado, utilizando o Raspberry Pi 4 como plataforma de testes. O dispositivo, amplamente empregado em soluções de prototipagem para sistemas embarcados e aplicações de IoT, representa uma base adequada para avaliar a viabilidade computacional da arquitetura híbrida WiSARD–SVR em cenários reais. A proposta será avaliada em termos de tempo de inferência, consumo de memória, taxa de acerto e estabilidade de funcionamento em comparação com soluções baseadas em deep learning, como o YOLOv8 — oferecendo uma análise direta entre complexidade algorítmica e desempenho prático.

Portanto, a originalidade deste trabalho reside não apenas na aplicação combinada de uma arquitetura não convencional de classificação com uma técnica consolidada de regressão para um problema relevante de visão computacional, mas também na ênfase na implementação real em hardware embarcado. Essa perspectiva busca demonstrar que modelos simples e eficientes podem competir, em determinados contextos, com soluções muito mais complexas e custosas, ampliando as possibilidades de aplicação da visão computacional em dispositivos de baixo custo e baixo consumo energético.

Diante desse contexto, este trabalho tem como objetivo investigar a viabilidade de modelos leves para a detecção de embarcações em imagens marítimas, com foco na combinação da arquitetura WiSARD para classificação e do SVR para regressão de coordenadas de caixas delimitadoras. A partir dessa proposta, formula-se o problema de pesquisa e as hipóteses de trabalho que norteiam esta dissertação.

## 1.1 Problema de Pesquisa

O problema central abordado nesta dissertação pode ser formulado nos seguintes termos:

*Como detectar embarcações em imagens marítimas de forma precisa, com baixo custo computacional, tempos de execução compatíveis com aplicações em tempo real e consumo energético reduzido, utilizando modelos que possam ser implementados em dispositivos de baixo custo, sem dependência de aceleração por hardware ou grandes volumes de dados para treinamento?*

O objetivo é viabilizar o uso de modelos de aprendizado de máquina embarcados em plataformas leves e acessíveis, como Raspberry Pi, voltadas a aplicações que exigem operação autônoma, eficiência energética e independência de infraestrutura computacional robusta.

## 1.2 Hipóteses do Trabalho

A hipótese central deste trabalho é que a combinação entre a rede WiSARD e a arquitetura SVR apresenta características promissoras para cenários de aprendizado de máquina na borda, permitindo realizar a detecção de embarcações de forma eficiente, precisa e energeticamente viável. A seguir, detalham-se as hipóteses específicas que sustentam essa proposição:

- **H1 — Eficácia da WiSARD para classificação:** A WiSARD, por meio de operações binárias simples e de sua estrutura baseada em memória associativa, viabiliza treinos rápidos e inferência com consumo mínimo de energia. Espera-se que, aplicada à tarefa de *classificação de embarcações*, essa arquitetura seja capaz de alcançar níveis satisfatórios de acurácia mantendo baixo custo computacional, características desejáveis em cenários de computação embarcada.
- **H2 — Adequação de métodos leves para regressão de *bounding boxes*:** Supõe-se que diferentes métodos de regressão computacionalmente leves, incluindo a *Regression* WiSARD, o kNN e o SVR, sejam capazes de estimar de forma consistente as coordenadas das caixas delimitadoras  $(x, y, w, h)$ . Espera-se que esses métodos apresentem desempenho competitivo em termos de erro de regressão e eficiência computacional, permitindo uma análise comparativa entre distintos paradigmas de aprendizado sob restrições de processamento e consumo energético.
- **H3 — Viabilidade do algoritmo de detecção híbrida em ambiente embarcado:** Espera-se que a integração da WiSARD para classificação com um módulo de regressão baseado em métodos leves resulte em um sistema híbrido capaz de alcançar bons níveis de acurácia na identificação das embarcações e baixos valores de erro médio absoluto (MAE) na regressão das caixas delimitadoras. Considera-se ainda que esse sistema possa ser implementado de forma eficiente em um dispositivo de baixo custo, como o Raspberry Pi 4, viabilizando uma solução embarcada para detecção de embarcações em tempo real sob restrições de energia e processamento.

A validação dessas hipóteses será conduzida por meio de uma estrutura experimental dividida em duas fases — computacional e embarcada — que permitirá comparar diretamente os modelos quanto a acurácia, tempo de execução, uso de memória e consumo energético, oferecendo uma visão abrangente sobre os *trade-offs* envolvidos na adoção de arquiteturas ultraleves para visão computacional em ambientes adversos.

O avanço da IoT tem transformado significativamente a forma como sistemas computacionais são concebidos e aplicados no mundo físico. A crescente conectividade entre sensores, atuadores e dispositivos inteligentes impulsiona a demanda por soluções descentralizadas, eficientes e autônomas, capazes de processar informações localmente — muitas vezes em tempo real e com severas restrições de energia, processamento e memória. Nesse contexto, a chamada Inteligência Artificial na borda (Edge AI) emerge como um paradigma essencial, possibilitando decisões rápidas e seguras diretamente nos dispositivos, sem depender da infraestrutura em nuvem.

O uso de algoritmos complexos, como modelos de deep learning, embora eficiente em termos de desempenho em cenários controlados, nem sempre é viável em sistemas embarcados com recursos limitados. Drones, veículos de superfície não tripulados (USVs), bóias oceânicas inteligentes e estações costeiras de vigilância são exemplos de plataformas onde o espaço físico, o consumo energético e a capacidade computacional são altamente restritos. A execução de modelos como YOLOv8 em tais dispositivos demanda soluções de aceleração específicas (como GPUs móveis ou TPUs) e estratégias complexas de otimização, que muitas vezes aumentam os custos e a complexidade do projeto.

Neste cenário, modelos como a WiSARD, com sua estrutura baseada em memórias RAM binárias, oferecem uma alternativa promissora. A ausência de operações matriciais intensivas e de processos iterativos de treinamento permite sua implementação em hardware modesto, como o Raspberry Pi, um dos dispositivos mais populares em prototipagem embarcada. Essa plataforma, dotada de capacidade computacional limitada, mas com ampla documentação e suporte, representa um excelente ambiente de validação para modelos que pretendem atuar em aplicações reais da IoT.

A baixa complexidade computacional, aliada à rapidez de aprendizado e à alta interpretabilidade das decisões tomadas pelo modelo WiSARD, reforça sua adequação para sistemas de monitoramento e detecção instalados em ambientes marítimos. Em contextos como a vigilância costeira ou o controle de tráfego portuário, a necessidade de respostas rápidas e consistentes, mesmo sob condições adversas ou interrupções de rede, torna impraticável a dependência de serviços remotos de processamento. Além disso, a possibilidade de realizar treinamentos e ajustes localmente, com conjuntos de dados reduzidos, amplia a adaptabilidade da solução a diferentes regiões geográficas, tipos de embarcações e cenários operacionais.

Portanto, a motivação para o desenvolvimento deste trabalho reside na convergência entre a necessidade real de sistemas de detecção mais acessíveis e a oportunidade de explorar modelos ainda pouco difundidos, mas com características tecnológicas altamente compatíveis com os desafios atuais da computação embarcada. A aplicação da WiSARD em visão computacional para IoT, especialmente no domínio marítimo, representa não apenas uma contribuição teórica, mas também uma proposta de alto impacto prático para soluções de baixo custo, alta autonomia e fácil integração em sistemas inteligentes distribuídos.

A metodologia adotada neste trabalho envolve o desenvolvimento, treinamento e avaliação de uma solução de detecção de embarcações baseada nas arquiteturas WiSARD e SVR. Inicialmente, será estruturado um *pipeline* de processamento de imagens, incluindo etapas de pré-processamento, extração de atributos e codificação binária utilizando codificação termométrica. A partir dessa representação, diferentes

modelos WiSARD serão treinados para executar as tarefas de classificação, como a detecção da presença de embarcações e a identificação de suas categorias. Para a tarefa de regressão das coordenadas de localização  $(x, y, w, h)$ , serão utilizadas quatro redes SVR independentes, cada uma dedicada à estimação de um dos parâmetros da *bounding box*.

A avaliação da proposta será conduzida em duas frentes: ambiente computacional padrão (notebook com recursos convencionais) e ambiente embarcado, utilizando o Raspberry Pi 4. Em ambos os contextos, serão analisados indicadores de desempenho mencionados abaixo. Os resultados serão comparados com os obtidos por modelos baseados em *deep learning*, particularmente o YOLOv8, permitindo uma análise crítica entre complexidade e eficácia em diferentes configurações.

Na avaliação experimental, serão consideradas métricas específicas de cada etapa da detecção. No caso da classificação, serão utilizadas precisão, revocação e F1-score, de modo a capturar diferentes aspectos do desempenho. Para a regressão das caixas delimitadoras, serão adotadas duas métricas complementares: o *Erro Absoluto Médio* (MAE), calculado separadamente para cada parâmetro  $(x, y, w, h)$ , permitindo identificar com clareza quais dimensões concentram os maiores erros; e a métrica *Intersection over Union* (IoU), que avalia a sobreposição global entre as caixas preditas e as reais. Além disso, serão medidos indicadores de eficiência computacional, incluindo tempo de inferência, tempo de treinamento e consumo energético, fundamentais para analisar a viabilidade prática da solução em dispositivos embarcados.

As principais contribuições desta dissertação podem ser sintetizadas em quatro pontos. Primeiro, propõe-se uma abordagem híbrida para detecção de embarcações em imagens marítimas que combina Redes Neurais Sem Peso, por meio da arquitetura WiSARD, com métodos leves de regressão para a estimação de caixas delimitadoras. Segundo, é realizada uma análise comparativa entre diferentes paradigmas de regressão computacionalmente eficientes, avaliando seu desempenho em termos de erro de regressão, tempo de execução e consumo energético. Terceiro, o trabalho investiga a viabilidade prática dessa abordagem em ambientes embarcados, validando sua execução em um dispositivo de baixo custo, como o Raspberry Pi 4, sob restrições reais de processamento e energia. Por fim, os resultados obtidos contribuem para a discussão sobre o uso de modelos não baseados em gradiente e arquiteturas leves como alternativas viáveis a métodos profundos tradicionais em aplicações de visão computacional embarcada.

A organização deste documento está estruturada da seguinte forma: no Capítulo 2, são discutidos os fundamentos teóricos necessários para o desenvolvimento deste trabalho, abrangendo redes neurais convolucionais, redes WiSARD e suas variações para tarefas de classificação e regressão, além do mecanismo de codificação termo-métrica e do uso de regressão via SVR. O Capítulo 3 apresenta uma revisão da

literatura, contemplando pesquisas relacionadas à detecção de embarcações e destacando o papel de modelos leves no contexto da Edge AI, de modo a situar o trabalho no estado da arte. No Capítulo 4, descreve-se a metodologia proposta, detalhando a concepção do *pipeline* híbrido, bem como o processo de construção, treinamento e implementação, complementado por uma descrição dos experimentos planejados. O Capítulo 5 concentra-se na apresentação e discussão dos resultados experimentais, trazendo análises comparativas do desempenho obtido frente a diferentes abordagens e evidenciando os principais achados do estudo. Finalmente, o Capítulo 6 reúne as conclusões, ressaltando as contribuições alcançadas, discutindo as limitações identificadas ao longo da pesquisa e indicando perspectivas promissoras para trabalhos futuros.

# Capítulo 2

## Conceitos Básicos

Este capítulo apresenta os fundamentos teóricos necessários para compreender o desenvolvimento deste trabalho. O objetivo é introduzir conceitos de visão computacional, redes neurais e modelos sem peso, além de técnicas de regressão aplicadas à predição de caixas delimitadoras em imagens. A discussão tem como foco estabelecer a base conceitual que sustenta a proposta metodológica, sem se aprofundar em detalhes de implementação, que são tratados nos capítulos posteriores.

### 2.1 Visão Computacional em Ambientes Marítimos

A visão computacional é uma área da inteligência artificial dedicada à interpretação e análise automática de imagens digitais. Seu avanço recente foi impulsionado pelo aumento da capacidade de processamento e pela popularização de arquiteturas de aprendizado profundo, que permitem a extração de representações complexas a partir de grandes volumes de dados LECUN *et al.* (2015).

No contexto marítimo, a aplicação da visão computacional apresenta desafios singulares. A superfície do mar é caracterizada por padrões dinâmicos e não estruturados, com presença de ondas, reflexos e variações de luminosidade ao longo do dia. Além disso, a mesma embarcação pode apresentar aparências distintas dependendo do ângulo de captura da imagem — vistas de popa, de través ou em diagonal revelam proporções e contornos diferentes, conforme figura 2.1, o que dificulta a generalização dos modelos de detecção. Outro fator crítico é a diversidade entre embarcações, que variam significativamente em proporção, escala e cor, aumentando a complexidade da tarefa ZHAO *et al.* (2024).



Figura 2.1: Diferentes perspectivas de um mesmo tipo de embarcação: vistas de través, frontal e traseira.

Essas dificuldades têm sido discutidas em diferentes estudos. Moosbauer et al. MOOSBAUER *et al.* (2019), ao proporem um *benchmark* para ambientes marítimos com base no *Singapore Maritime Dataset*, destacam o impacto dos reflexos da água e da iluminação variável no desempenho de detectores. ZHAO *et al.* (2024) reforçam esse ponto ao analisar imagens de sensoriamento remoto, apontando que embarcações podem estar densamente distribuídas, rotacionadas em diferentes ângulos e, muitas vezes, parcialmente ocluídas. De forma complementar, *surveys* recentes mostram que as condições ambientais adversas — como neblina, chuva e mar agitado — constituem barreiras recorrentes para algoritmos de visão computacional ZHANG *et al.* (2021); ZHAO *et al.* (2023).

Apesar desses obstáculos, o emprego de métodos de visão computacional em ambientes marítimos tem se mostrado essencial para aplicações de monitoramento, segurança e navegação autônoma. A detecção de embarcações, em particular, possibilita desde o apoio a operações de busca e salvamento até a fiscalização de atividades pesqueiras e comerciais, consolidando-se como um campo de grande relevância tanto acadêmica quanto prática ZHANG *et al.* (2021).

## 2.2 Redes Neurais em Visão Computacional

As redes neurais artificiais tornaram-se a base da visão computacional moderna devido à sua capacidade de aprender representações de alto nível diretamente a partir de dados brutos. A consolidação desse paradigma ocorreu com o avanço das redes neurais convolucionais (*Convolutional Neural Networks* — CNNs), inicialmente introduzidas por LeCun et al. LECUN *et al.* (1998) e popularizadas após o sucesso do AlexNet no ImageNet Challenge de 2012 KRIZHEVSKY *et al.* (2012). Esses modelos demonstraram desempenho sem precedentes em classificação de imagens, estabelecendo um novo marco no campo.

O funcionamento das CNNs baseia-se em filtros convolucionais capazes de capturar padrões locais, como bordas e texturas, que são progressivamente combinados ao longo das camadas para formar representações mais complexas. Formalmente, a operação de convolução entre uma imagem de entrada  $I$  e um filtro  $K$  pode ser descrita da seguinte forma:

$$S(i, j) = (I * K)(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n) \cdot K(m, n), \quad (2.1)$$

onde  $I(i, j)$  representa a intensidade do pixel na posição  $(i, j)$  da imagem de entrada,  $K(m, n)$  representa o peso do filtro na posição  $(m, n)$ , e  $S(i, j)$  é o valor resultante na posição  $(i, j)$  do mapa de saída. Essa operação é aplicada repetidamente em diferentes regiões da imagem, permitindo que o filtro aprenda a detectar padrões específicos, como arestas horizontais ou verticais.

Em imagens reais, geralmente representadas em múltiplos canais (por exemplo, RGB), a convolução é estendida de forma a considerar todos os canais de entrada. Nesse caso, cada filtro  $K$  possui uma profundidade igual ao número de canais  $C$ , e a operação é descrita como:

$$S(i, j) = \sum_{c=0}^{C-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I_c(i+m, j+n) \cdot K_c(m, n), \quad (2.2)$$

onde  $I_c$  e  $K_c$  correspondem, respectivamente, ao canal  $c$  da imagem de entrada e do filtro. Essa formulação permite que um único filtro combine informações provenientes dos diferentes canais, extraindo representações mais ricas.

Dois parâmetros adicionais influenciam diretamente o resultado da convolução: o *stride* e o *padding*. O *stride* define o número de pixels de deslocamento do filtro ao longo da imagem. Um *stride* igual a 1 implica que o filtro se move pixel a pixel, enquanto valores maiores reduzem a resolução espacial do mapa de saída, promovendo maior compactação da informação. O *padding*, por sua vez, corresponde à adição de valores artificiais (geralmente zeros) ao redor da borda da imagem de entrada. Esse

procedimento é utilizado para controlar as dimensões da saída, evitando a redução excessiva causada pela aplicação repetida de filtros e preservando informações das bordas.

Esses elementos matemáticos — canais, *stride* e *padding* — são fundamentais para compreender como CNNs conseguem equilibrar a extração de características detalhadas com a redução gradual da dimensionalidade. A hierarquia resultante de camadas convolucionais e operações auxiliares, como *pooling*, permite identificar desde estruturas simples até formas completas de objetos em uma cena, reduzindo a necessidade de engenharia manual de atributos. A introdução de arquiteturas mais profundas, como a ResNet HE *et al.* (2016), reforçou a importância das CNNs ao demonstrar que redes com centenas de camadas podem ser treinadas de forma estável e altamente eficaz.

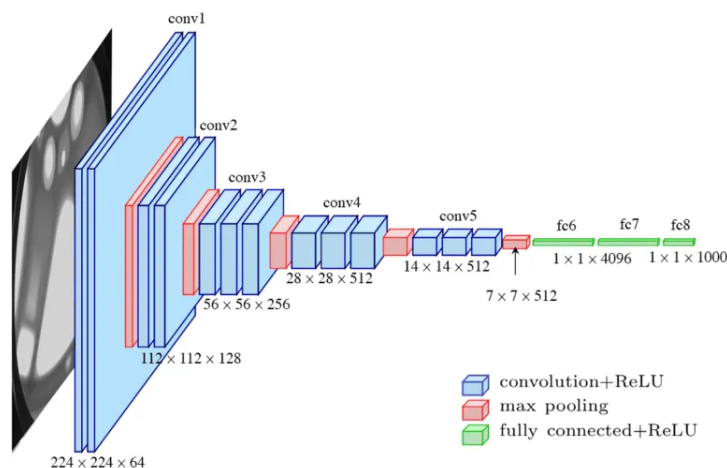


Figura 2.2: Arquitetura da rede VGG-16, ilustrando a progressão de camadas convolucionais, operações de *pooling* e camadas totalmente conectadas.

A Figura 2.2 ilustra de forma esquemática o funcionamento de uma rede neural convolucional profunda por meio da arquitetura VGG-16 SIMONYAN e ZISSERMAN (2014). A partir de uma imagem de entrada, as primeiras camadas convolucionais operam com filtros de pequena dimensão, responsáveis por extrair características locais simples, como bordas e variações de intensidade. À medida que a profundidade da rede aumenta, os mapas de características tornam-se progressivamente mais abstratos, representando padrões visuais de maior complexidade, como partes de objetos e estruturas semânticas. As operações de *max pooling*, intercaladas entre os blocos convolucionais, reduzem gradualmente a resolução espacial das ativações, contribuindo para a diminuição do custo computacional e para a incorporação de invariância a pequenas translações. Por fim, as camadas totalmente conectadas agregam as informações extraídas ao longo da hierarquia convolucional, produzindo uma representação global da imagem adequada para tarefas de classificação ou re-

gressão. Essa organização em estágios evidencia como os conceitos matemáticos da convolução se materializam em uma arquitetura hierárquica capaz de aprender representações visuais diretamente a partir dos dados.

Apesar de seu sucesso, as CNNs apresentam limitações significativas quando aplicadas em cenários restritos em termos de recursos computacionais, como dispositivos móveis ou sistemas embarcados. O elevado número de parâmetros e operações pode comprometer a viabilidade em aplicações de tempo real. Para contornar esse problema, foram propostas arquiteturas otimizadas, como a MobileNet HOWARD *et al.* (2017), que utiliza convoluções separáveis em profundidade para reduzir drasticamente o custo computacional mantendo níveis competitivos de acurácia. Trabalhos posteriores, como o MobileNetV2 e MobileNetV3, ampliaram essa ideia ao introduzir camadas de inversão residual e busca neural automatizada, consolidando a família MobileNet como referência em eficiência para visão computacional SANDLER *et al.* (2018); HOWARD *et al.* (2019).

Além das arquiteturas voltadas à eficiência, a detecção de objetos em tempo real se beneficiou de modelos como o YOLO (*You Only Look Once*), introduzido por Redmon et al. REDMON *et al.* (2016), que propôs um paradigma unificado de predição direta de caixas delimitadoras e classes em uma única etapa. Ao contrário dos detectores em duas fases, como o *Faster R-CNN*, o YOLO mostrou ser capaz de equilibrar precisão e velocidade, sendo amplamente empregado em aplicações práticas, inclusive em ambientes desafiadores como o marítimo.

O uso de CNNs como base para a visão computacional é, portanto, indispensável. No entanto, em diversos cenários, seu papel não é limitado à predição final, mas também à *extração de características*, em que as representações intermediárias (*feature maps*) de uma rede pré-treinada servem como entrada para modelos mais leves ou especializados. Essa estratégia, discutida com mais detalhes na subseção seguinte, permite explorar o poder de generalização das CNNs sem herdar integralmente seus custos computacionais, constituindo uma abordagem amplamente adotada em pesquisas recentes que visam soluções eficientes e adaptadas a ambientes específicos.

### 2.2.1 Extração de Características e *Transfer Learning*

Um dos papéis mais importantes das redes convolucionais em visão computacional é a capacidade de gerar representações intermediárias, chamadas de *mapas de características* (*feature maps*). Essas representações condensam informações relevantes da imagem original e podem ser utilizadas por outros algoritmos de aprendizado supervisionado, reduzindo a necessidade de treinar redes complexas de forma integral.

Formalmente, dado um conjunto de imagens de entrada  $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ , uma rede convolucional  $\phi(\cdot)$  pode ser interpretada como uma função que mapeia

cada imagem  $x_i \in \mathbb{R}^{H \times W \times C}$  para um vetor de características de dimensão reduzida  $z_i \in \mathbb{R}^d$ , tal que:

$$z_i = \phi(x_i; \theta), \quad (2.3)$$

onde  $\theta$  representa os parâmetros treinados da rede. O vetor  $z_i$  é então utilizado como entrada para modelos adicionais de classificação ou regressão. Esse processo corresponde ao paradigma conhecido como *transfer learning*, no qual o conhecimento adquirido em uma tarefa é reutilizado em um novo contexto PAN e YANG (2009).

Os *feature maps* resultam da aplicação sucessiva de convoluções, funções de ativação não-lineares (como ReLU) e operações de redução espacial (como *pooling*). Diferentes arquiteturas de CNN produzem representações com características distintas. Modelos como a MobileNet HOWARD *et al.* (2017); SANDLER *et al.* (2018) fornecem vetores compactos e eficientes, adequados para dispositivos de baixo consumo, enquanto arquiteturas mais profundas, como a VGG SIMONYAN e ZISSERMAN (2014) e a ResNet HE *et al.* (2016), geram representações altamente expressivas a partir de camadas convolucionais sucessivas. Mais recentemente, a EfficientNet TAN e LE (2019) introduziu um balanceamento sistemático entre profundidade, largura e resolução da rede, alcançando excelente desempenho com eficiência computacional superior às arquiteturas anteriores. O uso dessas representações intermediárias como entrada para algoritmos alternativos tem sido explorado com sucesso em diversas aplicações, incluindo ambientes marítimos, onde a extração de características robustas é fundamental diante da presença de ruídos e variações ambientais MOOS-BAUER *et al.* (2019).

A adoção de *transfer learning* oferece a vantagem de aproveitar conhecimento prévio adquirido em grandes bases de dados, como o ImageNet DENG *et al.* (2009), mitigando a necessidade de conjuntos de treino extensos em domínios específicos. Em termos práticos, a extração de características permite construir uma matriz de atributos  $\mathbf{Z} \in \mathbb{R}^{N \times d}$ , composta pelos vetores  $z_i$  de cada amostra, que pode ser utilizada como entrada para classificadores lineares, regressões ou outros modelos alternativos, como as redes sem pesos que serão discutidas na seção seguinte.

Assim, a etapa de extração de características desempenha papel central neste trabalho, pois possibilita equilibrar a capacidade de generalização das CNNs com a eficiência de algoritmos mais leves, viabilizando o desenvolvimento de soluções adequadas a cenários restritos em termos computacionais e energéticos.

## 2.3 Redes Neurais Sem Pesos (WNNs)

As Redes Neurais Sem Pesos (*Weightless Neural Networks* - WNNs) constituem uma classe de modelos de aprendizado de máquina cujo funcionamento é baseado em operações de escrita e leitura em memórias do tipo RAM (*Random Access Memory*), em vez do uso de pesos ajustados por algoritmos de otimização iterativa. Dentre as arquiteturas dessa família, destaca-se a WiSARD (*Wilkie, Stonham and Aleksander's Recognition Device*), proposta originalmente por Aleksander e Thomas na década de 1980 ALEKSANDER *et al.* (1984).

A WiSARD opera sobre entradas binárias, denominadas *retina*, que podem ser organizadas como vetores ou matrizes bidimensionais, como no caso de imagens em preto e branco. Quando os dados de entrada não são originalmente binários, técnicas de binarização ou codificação são empregadas para adequá-los ao modelo. O funcionamento da WiSARD é dividido em duas fases distintas: treinamento e classificação. Na fase de treinamento, informações são escritas nas memórias; na fase de classificação, essas informações são consultadas para produzir uma decisão.

A arquitetura da WiSARD é composta por quatro elementos principais: a retina, que representa o vetor binário de entrada; os neurônios, implementados como memórias RAM; os discriminadores, que consistem em conjuntos de neurônios associados a classes específicas; e um mecanismo de soma, responsável por medir o grau de similaridade entre uma entrada e os padrões previamente aprendidos por cada classe. O mapeamento entre os bits da retina e os endereços das memórias é realizado por meio de tuplas formadas a partir de subconjuntos do vetor de entrada, geralmente definidos de forma pseudoaleatória.

Durante o treinamento supervisionado, cada classe do problema é associada a um discriminador independente. Para uma amostra rotulada, os endereços definidos pelas tuplas são utilizados para acessar posições específicas das memórias do discriminador correspondente, onde informações são registradas. Esse processo ocorre de forma direta e não iterativa, caracterizando o aprendizado instantâneo (*one-shot learning*) da WiSARD.

Na fase de classificação, a entrada é apresentada a todos os discriminadores, que realizam leituras nas posições de memória correspondentes. As respostas dos neurônios são então agregadas por meio de um somador, produzindo uma medida de similaridade para cada classe. A classe associada ao maior valor agregado é selecionada como saída do modelo. Esse mecanismo equivale a um processo de votação distribuída entre os nós de memória.

Para reduzir ambiguidades e tornar a decisão mais robusta, pode-se empregar a técnica de *bleaching*, que introduz limiares progressivos para desconsiderar ativações de baixa frequência, enfatizando padrões mais representativos. Esse procedimento

contribui para a estabilidade da classificação sem aumento significativo do custo computacional.

As redes WiSARD apresentam vantagens importantes, como treinamento extremamente rápido, simplicidade estrutural e baixo custo computacional, características que as tornam especialmente adequadas para aplicações em ambientes de recursos restritos. Em contrapartida, sua dependência de entradas binárias exige etapas adicionais de pré-processamento. Ainda assim, a combinação entre eficiência, interpretabilidade e capacidade de aprendizado com poucos exemplos torna a WiSARD uma alternativa relevante para tarefas de classificação e reconhecimento de padrões, especialmente em cenários embarcados.

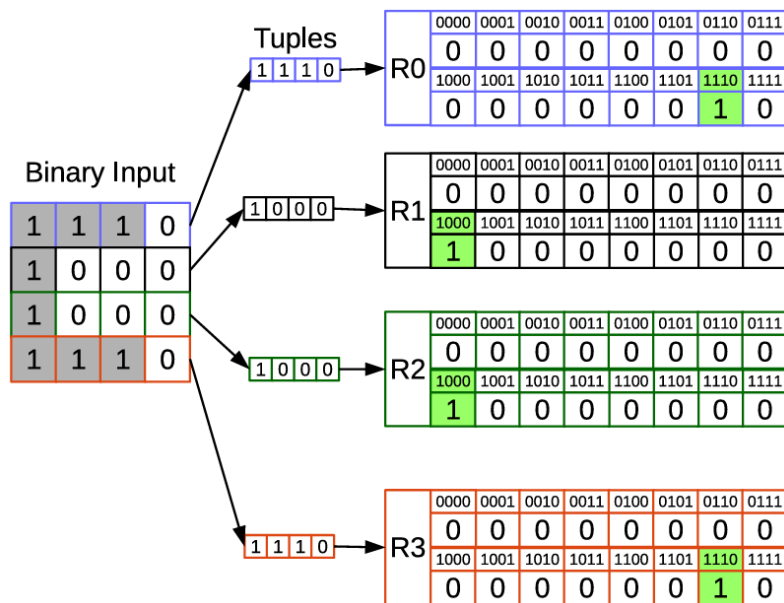


Figura 2.3: Exemplo de treinamento de uma rede WiSARD: a imagem é dividida em 4 tuplas de tamanho 4 e cada uma representa uma RAM. Cada vez que o respectivo endereço da tupla é acessado, seu valor na tabela vai de zero para um. Fonte: SANTIAGO *et al.* (2020)

### 2.3.1 WiSARD com *Bleaching*

Na arquitetura WiSARD original, todas as posições das memórias RAM são inicializadas com o valor zero, e durante o treinamento cada posição acessada recebe o valor um. Quando a rede é treinada com um grande número de amostras por classe, pode ocorrer a saturação dos discriminadores, isto é, um grande número de posições de memória passa a conter o valor um. Como consequência, diferentes discriminadores podem apresentar respostas idênticas durante a fase de classificação, resultando em empates que precisam ser resolvidos de forma arbitrária.

A técnica de *bleaching* foi proposta com o objetivo de reduzir a ocorrência des-

ses empates, introduzindo um mecanismo de desempate mais robusto baseado na frequência de ativação dos padrões GRIECO *et al.* (2010); CARVALHO *et al.* (2013). Para isso, o processo de treinamento da WiSARD é modificado de modo que, em vez de armazenar apenas o valor um nas posições acessadas das memórias, o valor armazenado seja incrementado a cada nova ocorrência daquele endereço. Dessa forma, cada posição de memória passa a representar o número de vezes em que um determinado padrão foi observado durante o treinamento.

Durante a fase de classificação, o comportamento dos neurônios também é ajustado. Em vez de retornar um valor igual a um sempre que uma posição de memória é acessada, o neurônio somente produz saída positiva caso o valor armazenado seja maior ou igual a um limiar denominado valor de *bleaching*, indicado por  $b$ . Inicialmente, o valor de  $b$  é definido como zero, o que torna o comportamento da rede equivalente ao da WiSARD original.

Caso ocorra empate entre os discriminadores, o valor de  $b$  é incrementado progressivamente, tornando o critério de ativação mais restritivo e privilegiando padrões que ocorreram com maior frequência durante o treinamento. Esse processo é repetido até que o empate seja resolvido, selecionando-se como resposta final a classe cujo discriminador apresente a maior soma de ativações.

O uso do *bleaching* permite melhorar a estabilidade da decisão e reduzir a influência de padrões pouco representativos ou ruidosos, sem introduzir etapas de otimização iterativa ou aumento significativo do custo computacional. Essa característica torna a técnica particularmente adequada para aplicações embarcadas, nas quais simplicidade e eficiência são requisitos fundamentais.

### 2.3.2 Regression WiSARD (ReWiSARD)

Embora a arquitetura WiSARD tenha sido originalmente concebida para tarefas de classificação, sua estrutura baseada em memória associativa permite extensões naturais para problemas de regressão, nos quais o objetivo é estimar valores contínuos. Nesse contexto, a *Regression WiSARD* (ReWiSARD) surge como uma adaptação da WiSARD clássica que preserva seus princípios fundamentais, ao mesmo tempo em que amplia seu domínio de aplicação para tarefas regressivas LUSQUINO FILHO *et al.* (2020).

A ReWiSARD mantém a organização básica da WiSARD, composta por retina binária, mapeamento em tuplas, neurônios implementados como memórias RAM e discriminadores independentes. A principal diferença reside na natureza da informação armazenada nas memórias. Enquanto na WiSARD de classificação as posições de memória armazenam valores binários ou contagens de ativações, na ReWiSARD essas posições passam a armazenar informações associadas a valores contínuos de

saída observados durante o treinamento.

Durante a fase de aprendizado, cada amostra de entrada binarizada é associada a um valor contínuo alvo. As posições de memória endereçadas pelas tuplas correspondentes à entrada recebem incrementos ou atualizações relacionadas a esse valor alvo, permitindo que cada neurônio acumule conhecimento estatístico sobre a variável a ser estimada. Esse processo ocorre de forma direta, sem o uso de algoritmos iterativos de otimização ou cálculo de gradientes, preservando o caráter não paramétrico e determinístico da arquitetura.

Na fase de inferência, a ReWiSARD realiza leituras nas posições de memória ativadas pelos padrões de entrada e combina as informações armazenadas para produzir uma estimativa contínua. A saída final é obtida por meio de estratégias simples de agregação, como médias ou somatórios normalizados, refletindo o comportamento estatístico aprendido durante o treinamento. Assim como na WiSARD clássica, a inferência baseia-se predominantemente em acessos à memória e operações aritméticas simples.

Do ponto de vista computacional, a ReWiSARD apresenta um compromisso distinto entre treinamento e inferência. O treinamento tende a ser mais custoso do que na WiSARD de classificação, devido ao armazenamento e à combinação de valores contínuos, enquanto a inferência permanece eficiente, baseada em operações simples de leitura e agregação. Essa característica torna a ReWiSARD particularmente atrativa para cenários em que a inferência contínua e o baixo consumo energético são requisitos prioritários, como em sistemas embarcados e aplicações de computação na borda.

Em síntese, a ReWiSARD estende a aplicabilidade das redes neurais sem pesos para problemas de regressão contínua, mantendo simplicidade estrutural, aprendizado não iterativo e eficiência computacional. Essas propriedades fazem da ReWiSARD uma alternativa relevante a métodos regressivos tradicionais em cenários de recursos restritos.

## 2.4 Support Vector Regression (SVR)

A regressão supervisionada tem como objetivo aprender uma função que relacione um conjunto de atributos de entrada a valores contínuos de saída. Entre os diversos métodos existentes, a *Support Vector Regression* (SVR) destaca-se como uma técnica robusta e amplamente estudada VAPNIK (1995); DRUCKER *et al.* (1997); SMOLA e SCHÖLKOPF (2004).

A SVR é uma extensão do arcabouço das Máquinas de Vetores de Suporte (SVM), originalmente propostas para problemas de classificação. Sua ideia central é encontrar uma função de regressão  $f(x)$  que aproxime os valores reais  $y$ , permitindo

uma margem de tolerância  $\varepsilon$  dentro da qual desvios não são penalizados. Essa margem define um “tubo de insensibilidade” ao redor da função de regressão, ilustrando a noção de que pequenos erros são aceitáveis desde que o modelo capture de maneira geral a relação entre as variáveis.

Matematicamente, a função de regressão na SVR é definida como

$$f(x) = \langle w, \phi(x) \rangle + b, \quad (2.4)$$

onde  $x \in \mathbb{R}^d$  representa o vetor de entrada,  $\phi(x)$  é uma transformação (possivelmente não linear) do espaço de entrada para um espaço de características de maior dimensão,  $w$  é o vetor de pesos e  $b$  o viés. A função de perda utilizada é a chamada *epsilon-insensitive loss*, expressa como

$$L_\varepsilon(y, f(x)) = \max\{0, |y - f(x)| - \varepsilon\}. \quad (2.5)$$

Essa formulação significa que apenas erros acima de um limiar  $\varepsilon$  são considerados no ajuste do modelo, tornando-o mais robusto a ruídos. O parâmetro  $C > 0$  controla o equilíbrio entre a largura da margem e a penalização de erros maiores que  $\varepsilon$ . Como resultado, apenas algumas amostras de treino, chamadas de *vetores de suporte*, influenciam diretamente a forma final da função, o que confere à SVR um caráter parcimonioso.

Um aspecto importante da SVR é a possibilidade de incorporar funções *kernel*, que permitem lidar com relações não lineares entre atributos de entrada e saída sem precisar calcular explicitamente a transformação  $\phi(x)$ . Um dos *kernels* mais utilizados é o radial (*Radial Basis Function* — RBF), definido como

$$K(x, x') = \exp(-\gamma \|x - x'\|^2), \quad (2.6)$$

onde  $\gamma$  é um parâmetro que controla a influência de cada amostra. Esse recurso amplia a flexibilidade do modelo, tornando-o capaz de capturar dependências complexas de forma eficiente.

Portanto, a SVR fornece um equilíbrio interessante entre robustez estatística, capacidade de generalização e flexibilidade não linear, sendo adequada para uma ampla variedade de problemas de regressão.

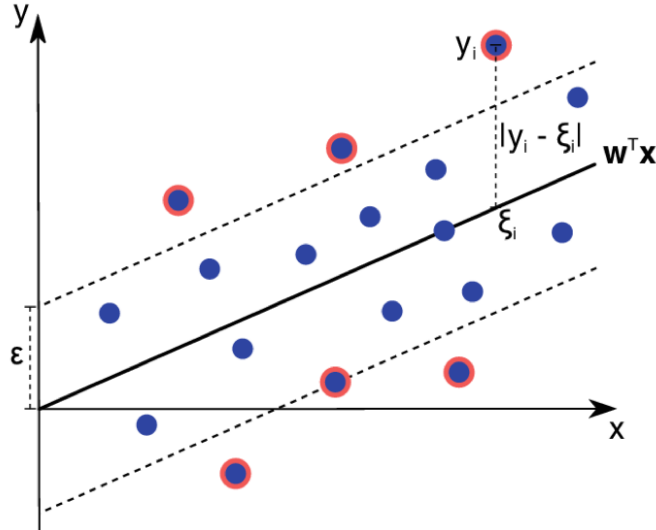


Figura 2.4: Exemplo de regressão por vetores de suporte (SVR). Ilustração de uma função de regressão SVR representada por  $w^T x$ . O tubo  $\epsilon$ -insensível em torno da função é indicado em cinza.  $\xi_i = w^T x_i$  corresponde ao valor previsto para a amostra  $x_i$ , enquanto  $y_i$  representa o valor real. Os vetores de suporte estão destacados com borda vermelha. Fonte: ROSENBAUM *et al.* (2013)

## 2.5 k-Nearest Neighbors (kNN)

O *k-Nearest Neighbors* (kNN) é um método de aprendizado supervisionado amplamente utilizado tanto em tarefas de classificação quanto de regressão, caracterizado por sua simplicidade conceitual e natureza não paramétrica ???. Diferentemente de modelos baseados em otimização, o kNN não constrói explicitamente um modelo durante a fase de treinamento, baseando suas previsões diretamente nas amostras do conjunto de treino.

Na regressão por kNN, a previsão de um valor contínuo para uma nova amostra é obtida a partir dos valores associados às  $k$  amostras mais próximas no espaço de características. Dado um vetor de entrada  $x \in \mathbb{R}^d$ , o algoritmo identifica os  $k$  vizinhos mais próximos segundo uma métrica de distância, sendo a distância Euclidiana a mais comumente empregada, definida como

$$d(x, x_i) = \|x - x_i\|_2. \quad (2.7)$$

Uma vez identificados os vizinhos mais próximos, a saída regressiva é calculada como a média dos valores associados a essas amostras, ou, de forma mais geral, como

uma média ponderada pelas distâncias:

$$\hat{y}(x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} y_i, \quad (2.8)$$

onde  $\mathcal{N}_k(x)$  representa o conjunto dos  $k$  vizinhos mais próximos de  $x$  e  $y_i$  o valor real associado à amostra  $x_i$ . Em variantes ponderadas, amostras mais próximas recebem maior peso na estimativa final, reduzindo a influência de vizinhos mais distantes.

O principal hiperparâmetro do método é o número de vizinhos  $k$ , que controla o compromisso entre viés e variância do modelo. Valores pequenos de  $k$  tendem a capturar melhor variações locais, mas podem tornar o modelo sensível a ruídos, enquanto valores maiores promovem maior suavização da função de regressão, ao custo de possível perda de detalhes.

Uma característica fundamental do kNN é que seu custo computacional concentra-se na fase de inferência, uma vez que o cálculo das distâncias deve ser realizado em relação a todo o conjunto de treinamento. Por outro lado, o treinamento é praticamente imediato, consistindo apenas no armazenamento das amostras. Essa propriedade torna o kNN atrativo em cenários nos quais o treinamento é frequente ou o volume de dados é moderado, mas pode limitar sua aplicação em ambientes de recursos restritos quando a inferência precisa ser realizada continuamente.

Em síntese, o kNN oferece uma abordagem intuitiva e flexível para regressão, servindo frequentemente como método de referência ou alternativa leve em problemas nos quais a simplicidade do modelo e a ausência de fase de treinamento explícita são desejáveis.

## 2.6 Resumo do Capítulo

Neste capítulo foram apresentados os conceitos fundamentais que embasam o desenvolvimento desta dissertação. Inicialmente, discutiram-se os desafios da visão computacional em ambientes marítimos, enfatizando as dificuldades impostas por reflexos, variações de luminosidade, condições climáticas adversas e pela diversidade de perfis e proporções das embarcações.

Na sequência, revisaram-se as redes neurais convolucionais, destacando sua importância no avanço da área e seu papel como extratoras de características em representações intermediárias. Foram mencionadas arquiteturas como MobileNet, ResNet e EfficientNet, que se diferenciam em termos de profundidade, eficiência e capacidade de generalização, constituindo a base para modelos mais leves e especializados.

Além disso, foi introduzido o conceito de WNNs, com foco na arquitetura Wi-

SARD, que se destaca pelo aprendizado instantâneo e baixo custo computacional, embora dependa de representações binárias adequadas. Por fim, abordou-se a *Support Vector Regression* (SVR), uma extensão das Máquinas de Vetores de Suporte para regressão de valores contínuos, que equilibra robustez, generalização e flexibilidade por meio do uso de funções *kernel*.

Esses conceitos fornecem a base teórica necessária para compreender a proposta metodológica apresentada nos capítulos seguintes, além de situar este trabalho dentro do contexto mais amplo da visão computacional aplicada ao ambiente marítimo.

# Capítulo 3

## Trabalhos Relacionados

Este capítulo está estruturado em quatro eixos principais. Primeiro, são discutidos os estudos sobre a rede WiSARD. Em seguida, apresentam-se as aplicações da WiSARD em visão computacional, destacando sua utilização em tarefas de reconhecimento de padrões visuais. O terceiro eixo aborda o uso da *Support Vector Regression* (SVR) em problemas de visão computacional, enquanto o quarto e último concentra-se em abordagens baseadas em redes neurais profundas aplicadas à detecção de embarcações. Essa organização permite situar a presente pesquisa dentro do estado da arte, identificando lacunas e oportunidades que motivam a proposta metodológica desenvolvida.

### 3.1 Introdução

O avanço científico em visão computacional, aprendizado de máquina e sistemas embarcados tem resultado em uma ampla variedade de métodos aplicados à detecção e ao reconhecimento de padrões. Compreender esse panorama é essencial para situar a presente dissertação no contexto acadêmico mais amplo, permitindo destacar contribuições existentes, reconhecer limitações recorrentes e estabelecer os fundamentos que sustentam as escolhas metodológicas adotadas. A partir desta perspectiva, apresentam-se a seguir os trabalhos mais relevantes que dialogam diretamente com este estudo.

### 3.2 WiSARD e Redes Neurais Sem Peso

O modelo WiSARD (Wilkie, Stonham and Aleksander Recognition Device) é um exemplo clássico de Rede Neural sem pesos (WNN), proposto na década de 1980 como alternativa às redes convencionais baseadas em pesos ajustados por retropropagação ALEKSANDER *et al.* (1984). Sua arquitetura é composta por nós de

memória RAM que armazenam associações entre subcadeias binárias do vetor de entrada e classes de saída. O processo de classificação é realizado por meio de votação entre os valores recuperados nessas memórias, permitindo aprendizado rápido e de baixa complexidade computacional.

Nos últimos anos, diferentes variações e implementações da WiSARD foram propostas, buscando adaptar o modelo a contextos modernos de aprendizado de máquina. De Gregorio e Giordano DE GREGORIO e GIORDANO (2016) apresentaram análises comparativas do classificador WiSARD em múltiplos conjuntos de dados, discutindo suas vantagens e limitações em relação a técnicas tradicionais. Outras propostas avançaram em direção a otimizações de hardware. O LogicWiSARD, por exemplo, substitui memórias convencionais por implementações lógicas, reduzindo o consumo de energia em dispositivos reconfiguráveis MIRANDA *et al.* (2022). Em paralelo, a arquitetura BTHOWeN introduziu filtros de Bloom contadores e codificação em termômetro, ampliando a expressividade do modelo e tornando-o competitivo em tarefas de classificação de imagens SUSSKIND *et al.* (2022). Além disso, iniciativas como a biblioteca wisardpkg LIMA FILHO *et al.* (2020) têm facilitado a experimentação prática, disponibilizando implementações em C++ e Python.

No domínio da visão computacional, a WiSARD tem sido utilizada em diferentes aplicações. Trabalhos iniciais exploraram o reconhecimento de caracteres manuscritos, aproveitando a compatibilidade entre a binarização das imagens e a estrutura do modelo. Outras investigações aplicaram a WiSARD ao reconhecimento facial e à classificação de imagens médicas, como radiografias e tomografias, ressaltando sua rapidez de treinamento e interpretabilidade. Mais recentemente, Milhomem *et al.* MILHOMEM *et al.* (2019) combinaram técnicas de transferência de aprendizado com o modelo WiSARD para detectar defeitos em pavimentos de asfalto. Nesse estudo, foram obtidos resultados de acurácia em torno de 85 por cento, demonstrando a viabilidade do modelo em cenários com imagens reais e complexas.

Apesar dessas contribuições, a aplicação da WiSARD em visão computacional permanece limitada quando comparada à predominância das redes neurais profundas. A literatura aponta que a simplicidade da representação binária pode restringir sua capacidade em contextos visuais mais desafiadores. Ainda assim, os trabalhos existentes indicam que a hibridização com métodos de extração de características mais robustos, como CNNs, pode ampliar seu potencial. Essa combinação surge como uma linha promissora de pesquisa, explorada nesta dissertação para lidar com os desafios específicos da detecção de embarcações em ambientes marítimos.

### 3.3 Support Vector Regression em Visão Computacional

A *Support Vector Regression* (SVR) é uma técnica derivada das Máquinas de Vetores de Suporte (SVM), originalmente propostas para classificação binária. Enquanto as SVMs buscam encontrar um hiperplano que maximize a separação entre classes, a SVR tem como objetivo aproximar uma função contínua que relacione atributos de entrada e valores de saída. Para isso, utiliza a função de perda epsilon-insensitive, que define um intervalo de tolerância em torno da função de regressão no qual desvios não são penalizados SMOLA e SCHÖLKOPF (2004). Esse mecanismo torna o modelo mais robusto a ruídos e garante maior capacidade de generalização em contextos onde pequenas variações não comprometem a qualidade da predição.

Uma característica central da SVR é o uso de vetores de suporte, isto é, apenas um subconjunto das amostras de treino influencia diretamente a solução final. Esse princípio reduz a complexidade do modelo e favorece sua aplicação em cenários de alta dimensionalidade. Além disso, a introdução de funções kernel possibilita modelar relações não lineares sem necessidade de transformar explicitamente os dados para espaços de maior dimensão. O kernel radial (RBF), definido como  $K(x, x') = \exp(-\gamma\|x - x'\|^2)$ , é amplamente utilizado em visão computacional por sua capacidade de capturar padrões complexos em imagens SMOLA e SCHÖLKOPF (2004).

A SVR tem sido aplicada com sucesso em diferentes problemas visuais. Agarwal e Triggs AGARWAL e TRIGGS (2006) empregaram regressões SVR para estimar a pose humana a partir de descritores de imagem, explorando sua eficiência em prever ângulos articulares de forma contínua. Guo e Fu GUO e FU (2008) aplicaram a técnica à estimativa de idade em faces humanas, obtendo resultados competitivos com descritores inspirados em características biológicas. Na área médica, a SVR tem sido utilizada tanto para regressão de medidas morfométricas a partir de imagens radiográficas quanto em análises hiperespectrais, como no trabalho de Tuia et al. TUIA *et al.* (2011), que demonstraram sua eficiência na predição de parâmetros biofísicos em sensoriamento remoto de alta dimensionalidade.

Outros estudos ampliaram a aplicação da SVR para tarefas de reconstrução de superfícies tridimensionais. Wang e Sun WANG e SUN (2005) apresentaram um método baseado em SVR para reconstrução de superfícies de forma livre a partir de dados de pontos, demonstrando que a técnica é capaz de lidar com representações geométricas contínuas de maneira eficaz. Trabalhos em análise de expressões faciais também mostram que a SVR pode prever intensidades de emoções em escalas contínuas, ao invés de tratá-las apenas como categorias discretas, reforçando sua flexibilidade em tarefas de regressão visual.

As vantagens da SVR em visão computacional incluem a robustez frente a conjuntos de dados menores, a capacidade de generalizar bem em alta dimensionalidade e a interpretabilidade associada à noção de vetores de suporte. Entretanto, limitações também são relatadas, como a dificuldade em lidar com bases de dados extremamente grandes, dado que o custo de treinamento cresce de forma quadrática com o número de amostras. Nesse sentido, modelos profundos têm superado a SVR em tarefas com grande disponibilidade de dados rotulados, mas a SVR permanece uma escolha competitiva em cenários de dados restritos ou quando se busca equilíbrio entre desempenho e custo computacional.

### 3.4 Detecção de Embarcações com Redes Neurais Profundas

A detecção de embarcações utilizando redes neurais profundas tem se intensificado nos últimos anos, aproveitando tanto imagens ópticas quanto imagens de Radar de Abertura Sintética (SAR). Esses modelos têm se apoiado em arquiteturas como Faster R-CNN, YOLO, SSD, e variantes com também detecção rotacionada, detecção em múltiplas escalas e melhorias de *backbone*.

Zhao et al. ZHAO *et al.* (2024) apresentam uma revisão sistemática de métodos de detecção de navios em imagens ópticas de sensoriamento remoto, classificando os desafios — variação de escala, orientação, aspecto, fundos complexos — e comparando diferentes soluções baseadas em CNNs e Transformers. Outra contribuição importante é *R-CNN-Based Ship Detection from High Resolution Remote Sensing Imagery* de Zhang et al. ZHANG *et al.* (2019), que propõe uma Faster R-CNN modificada com VGG-16 e *pooling* de múltiplas resoluções para tratar navios pequenos e grupos de navios próximos uns aos outros.

Além disso, Li LI *et al.* (2021) oferece métodos práticos de classificação e detecção em imagens ópticas, com avaliações robustas em *datasets* públicos. Trabalhos mais recentes focam em eficiência e detecção de navios pequenos. Por exemplo, o YOLO-ssboat WANG *et al.* (2025b) propõe melhorias para detectar embarcações pequenas em ambientes marítimos complexos, integrando módulos de convolução deformável, pirâmides de múltiplas escalas, atenção multi-escala e fusão de cabeças de detecção para aumentar a precisão em objetos pequenos, mesmo quando há interferência de esteiras de onda no mar.

Modelos leves também surgem como alternativa. O SDNet LIU *et al.* (2025) propõe uma rede baseada em super-resolução e detalhamento fino para melhorar a detecção com custo computacional reduzido.

Outra linha com radar SAR é o trabalho de Xiao et al. XIAO *et al.* (2022), que

propõem a detecção de navios baseada em Swin Transformer e *Feature Enhancement Feature Pyramid Network* (FEFPN), combinando Transformers com pirâmides de características para capturar dependências longas e melhorar desempenho em navios de pequeno porte em imagens SAR do *dataset* SSDD.

Também há abordagens voltadas para cenários reais ou variados. Betti et al. BETTI *et al.* (2020) apresentam um conjunto de mais de 56 mil imagens de embarcações e múltiplas categorias (navio de carga, navio naval, rebocador etc.), demonstrando que modelos YOLOv3 podem alcançar precisão elevada ( $mAP/IoU \approx 0.5$ ) em condições variadas.

Em conjunto, esses trabalhos mostram que:

- é possível obter desempenho muito bom com modelos profundos, mesmo para objetos pequenos ou em fundos marítimos complexos;
- melhores resultados são alcançados quando se usam *features* multi-escala, *backbones* mais robustos, técnicas de fusão de cabeças de detecção e uso de *bounding boxes* rotacionadas;
- modelos leves ou adaptados (por exemplo, com camadas mais simples ou estruturas otimizadas) têm sido propostos, mas muitas vezes há *trade-offs* entre acurácia e eficiência;
- há um crescente interesse em lidar com condições adversas de fundo marítimo, ruído (ondas, reflexos, neblina), escala variável e navios pequenos.

Observa-se que, apesar dos avanços recentes, a literatura ainda carece de abordagens que conciliem desempenho competitivo com eficiência energética e baixo custo computacional, especialmente em cenários de *Edge AI* e sistemas embarcados. Nesse sentido, o presente trabalho se diferencia ao propor um pipeline alternativo, baseado em Redes Neurais sem pesos combinadas com regressão por vetores de suporte, buscando preencher essa lacuna e ampliar as possibilidades de detecção de embarcações em ambientes reais.

# Capítulo 4

## Estratégia de Detecção de Objeto Único em Borda

Este capítulo apresenta a proposta deste trabalho, denominada SOD-Edge (*Single Object Detection for Edge Computing*), com ênfase no algoritmo desenvolvido e na definição das estruturas de dados utilizadas. Primeiramente, descreve-se a visão geral da solução proposta e seus princípios. Em seguida, é apresentada a formulação do problema e a estrutura dos dados que compõem a solução. Por fim, são detalhadas as etapas do *pipeline* implementado, desde o processamento inicial das imagens até a predição final.

### 4.1 Visão Geral da Proposta

Esta dissertação propõe o desenvolvimento de um sistema leve e eficiente, denominado SOD-Edge, com baixo custo computacional, tempos de execução compatíveis com aplicações em tempo real e consumo energético reduzido, voltado à detecção de embarcações em ambientes de borda. A proposta fundamenta-se no uso de Redes Neurais Sem Peso, mais especificamente a arquitetura WiSARD, em combinação com diferentes abordagens de regressão para a predição das *bounding boxes*.

O trabalho propõe a utilização da WiSARD tradicional para a tarefa de classificação das embarcações e a aplicação de modelos de regressão independentes para a estimativa das coordenadas e dimensões da *bounding box*, permitindo não apenas a identificação do tipo de embarcação, mas também sua localização espacial precisa. Nesse contexto, a proposta passa a considerar três abordagens distintas para regressão: *Support Vector Regression* (SVR), *k-Nearest Neighbors* (kNN) e uma versão estendida da própria arquitetura WiSARD adaptada para regressão, denominada *Regression WiSARD* (ReWiSARD). Essa formulação amplia o escopo da solução, permitindo a análise comparativa entre diferentes paradigmas de regressão sob as

mesmas restrições computacionais.

O diferencial da abordagem proposta está na utilização de um modelo sem pesos, cuja estrutura é baseada em memórias associativas, o que elimina a necessidade de algoritmos de retropropagação tradicionais. Diferentemente das redes neurais convencionais, como as CNNs, que ajustam funções de ativação e pesos por meio de processos iterativos de otimização, a WiSARD realiza o aprendizado de forma direta e não iterativa, armazenando os padrões de entrada em posições de memória específicas. Durante a fase de inferência, a rede compara os padrões armazenados com as novas entradas, contabilizando o número de coincidências em cada discriminador para determinar a classe mais provável.

Essa característica torna o processo de aprendizado mais rápido, interpretável e energeticamente eficiente, além de eliminar completamente a necessidade de cálculo de gradientes e atualização de pesos — aspectos centrais em arquiteturas de aprendizado profundo. Esse comportamento também viabiliza um paradigma de *one-shot learning*, no qual os dados de treino são apresentados uma única vez ao modelo, proporcionando tempos de treinamento quase instantâneos e consumo energético reduzido, atributos altamente desejáveis em aplicações embarcadas.

No que diz respeito à tarefa de regressão das *bounding boxes*, a proposta explora diferentes estratégias. O uso do SVR é motivado por sua capacidade de modelar relações não lineares em espaços de alta dimensão, com bom controle da complexidade do modelo. O kNN é considerado por sua simplicidade computacional e comportamento não paramétrico, sendo especialmente atrativo em cenários de dados limitados. Já a ReWiSARD representa uma extensão natural do paradigma sem pesos para tarefas de regressão, preservando as vantagens de aprendizado direto e baixo custo computacional. Embora técnicas de regressão já tenham sido exploradas em visão computacional para problemas como estimativa de pose e análise de atributos visuais, seu uso em detecção de objetos, especialmente em arquiteturas leves voltadas à borda, permanece pouco explorado. Dessa forma, a combinação entre WiSARD e múltiplos regressores constitui o núcleo da proposta SOD-Edge para sistemas de detecção em ambientes restritos.

O algoritmo desenvolvido será composto, conforme ilustrado na Figura 4.1, pelas seguintes etapas:

- **Pré-processamento e extração de atributos:** As imagens de entrada são redimensionadas e processadas por uma CNN leve e pré-treinada, selecionada por seu equilíbrio entre desempenho e eficiência, com extração de mapas de características em dois pontos distintos da rede;
- **Módulo de classificação:** Uma instância WiSARD tradicional é responsável por identificar o tipo de embarcação entre as sete categorias previstas na base

de dados;

- **Módulo de regressão:** Modelos de regressão independentes — SVR, kNN ou ReWiSARD — são utilizados para prever, respectivamente, as coordenadas  $(x, y)$  e as dimensões  $(w, h)$  da *bounding box* que delimita o objeto detectado.

A avaliação da proposta será realizada em duas fases experimentais, visando caracterizar o desempenho tanto em ambiente computacional robusto quanto em sistema embarcado de baixo custo:

1. **Ambiente computacional:** Conduzido na plataforma Google Colab Pro com suporte a GPU, onde será realizada a comparação entre o modelo proposto e a arquitetura YOLOv8, considerada um dos principais representantes do estado da arte;
2. **Ambiente embarcado:** Realizado em um Raspberry Pi 4 com 8 GB de RAM, com o objetivo de verificar a viabilidade da execução da solução SOD-Edge em um cenário de recursos computacionais e energéticos limitados.

Ao final, espera-se demonstrar que modelos simples e energeticamente eficientes constituem alternativas viáveis a soluções baseadas em aprendizado profundo em contextos onde a infraestrutura computacional é restrita, sem comprometer significativamente a precisão e a capacidade de generalização. Com isso, pretende-se contribuir para o avanço de soluções de visão computacional embarcada que sejam acessíveis, escaláveis e sustentáveis.

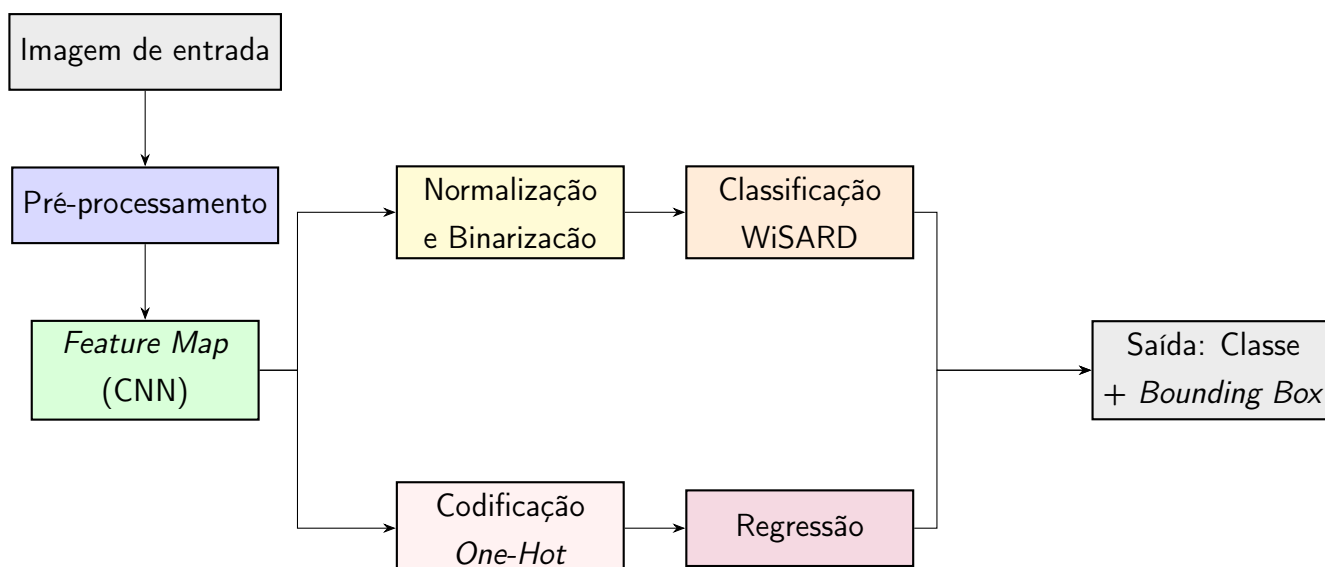


Figura 4.1: Pipeline proposto para detecção de embarcações. A imagem de entrada passa por pré-processamento e extração de características com CNN; as *features* seguem dois fluxos: (i) binarização e classificação pela WiSARD e (ii) codificação e regressão das *bounding boxes*. Ambos convergem para uma saída única contendo a classe e a localização da embarcação.

## 4.2 Formulação do Problema

A detecção automática de embarcações em imagens marítimas desempenha um papel estratégico, tático e até operacional em diversas áreas, incluindo vigilância costeira, proteção ambiental e logística no porto. No entanto, a implementação desses sistemas em contextos embarcados, como drones, boias inteligentes e embarcações autônomas, enfrenta desafios significativos, especialmente relacionados com o consumo de energia, a limitação de hardware e a necessidade de respostas em tempo real.

As abordagens tradicionais de detecção baseadas em aprendizado profundo, como as arquiteturas da série YOLO, demandam considerável poder computacional para alcançar bons resultados, tanto no treinamento quanto na inferência. Apesar dos avanços em técnicas de compressão e modelos leves, como MobileNet HOWARD *et al.* (2017) HOWARD *et al.* (2017) HOWARD *et al.* (2019) e YOLO-Nano WONG *et al.* (2019), essas soluções ainda requerem dispositivos com suporte a GPU ou aceleradores dedicados, o que pode inviabilizar sua aplicação em sistemas de baixo custo e com recursos limitados.

Neste contexto, torna-se necessária uma reavaliação das arquiteturas atualmente empregadas na detecção de objetos, especialmente quando se busca levar a inteligên-

cia artificial até a borda (*Edge AI*) de forma sustentável e operacionalmente viável. Essa necessidade decorre das limitações inerentes aos dispositivos embarcados, que impõem restrições de processamento, memória e consumo energético. Surge, portanto, a demanda por abordagens alternativas capazes de combinar:

- Reduzido tempo de inferência e consumo de energia durante o processamento;
- Estruturas de treinamento simples e rápidas, que dispensem processos iterativos complexos e longos tempos de otimização;
- Capacidade de inferência em tempo real, mesmo na ausência de aceleração por hardware especializado (como GPUs ou TPUs);
- Compatibilidade com dispositivos de baixo custo e operação autônoma, assegurando aplicabilidade em cenários embarcados e remotos.

## 4.3 Estruturas de Dados

A estruturação dos dados utilizados nesta dissertação segue um fluxo modular, orientado à compatibilidade com modelos leves de inferência binária, como a WiSARD, e ao reaproveitamento para comparações com modelos baseados em aprendizado profundo, como o YOLOv8. O pipeline proposto contempla desde a preparação das imagens brutas até a codificação final dos atributos e rótulos.

### 4.3.1 Imagem de entrada e pré-processamento

O conjunto de dados é composto por imagens RGB anotadas manualmente, contendo embarcações em variados contextos marítimos. Como etapa inicial, as imagens passam por um processo de redimensionamento para  $224 \times 224$  pixels.

Essa escolha de resolução representa um compromisso entre detalhamento visual e economia de recursos computacionais. Resoluções maiores demandariam maior tempo de inferência e memória, o que prejudicaria o desempenho em dispositivos embarcados. Já resoluções muito baixas comprometem a qualidade da extração de atributos e a precisão dos modelos de regressão. A resolução  $224 \times 224$  é amplamente utilizada em aplicações com redes convolucionais leves KUZMIC e RUDOLPH (2021) e se mostrou adequada durante os testes preliminares.

### 4.3.2 Extração de características com CNN leve

As imagens redimensionadas são processadas por uma MobileNetV3 Small HOWARD *et al.* (2019), empregada como *backbone* da solução para extração de mapas de características. Essa rede foi previamente treinada no conjunto de dados

ImageNet-1K, composto por mais de 1,2 milhão de imagens de 1.000 categorias, o que garante a extração de representações visuais robustas e generalizáveis. A escolha pela MobileNetV3 Small se justifica pela sua excelente relação entre desempenho e eficiência computacional, aspecto fundamental em aplicações destinadas a sistemas embarcados e dispositivos de baixa capacidade de processamento. Essa arquitetura foi projetada com foco explícito em inferência rápida e consumo reduzido de energia, alcançando resultados competitivos em tarefas de classificação e extração de características visuais, mesmo sob restrições de hardware.

A MobileNetV3 Small combina avanços introduzidos nas versões anteriores com otimizações específicas. Sua estrutura baseia-se em camadas convolucionais com separação de profundidade (*depthwise separable convolutions*), o que reduz drasticamente o número de parâmetros e operações de multiplicação-acumulação (MACs) quando comparada a redes convolucionais tradicionais. Além disso, a variante *Small* foi desenhada para cenários de baixa latência, priorizando convoluções com menor número de canais.

Foram exploradas duas saídas distintas dessa rede, cada uma com características complementares:

- **Saída intermediária (features.9):** corresponde ao nono bloco convolucional da rede, que mantém resolução espacial relativamente alta em relação às camadas posteriores. O formato resultante é  $96 \times 7 \times 7$ , totalizando 4.704 valores. Essa representação preserva informações locais de posicionamento e bordas, sendo adequada para a tarefa de regressão. Ela é empregada pelos quatro modelos responsáveis pela predição das coordenadas  $(x, y)$  e dimensões  $(w, h)$  das *bounding boxes*;
- **Saída profunda (features.12, avgpool):** refere-se ao penúltimo bloco convolucional (features.12) seguido pela camada de *average pooling global* (avgpool). Essa operação reduz a dimensão espacial da saída para  $1 \times 1$ , resultando em um vetor compacto de 576 valores ( $576 \times 1 \times 1$ ). Essa representação concentra maior abstração semântica, capturando padrões de alto nível (como formas globais e texturas), sendo ideal para alimentar a rede WiSARD no módulo de classificação.

Essa separação entre representações espaciais e semânticas permite que cada subtarefa se beneficie do tipo de informação mais apropriado: a regressão depende de detalhes espaciais finos, enquanto a classificação exige vetores mais compactos e discriminativos.

### 4.3.3 Normalização e codificação dos atributos

Após a extração dos mapas de características é realizada a normalização dos dados. Para tornar o primeiro mapa de vetores extraídos compatível com a arquitetura WiSARD, é necessário converter os valores contínuos em representações binárias. Para isso, é empregada a técnica de codificação termométrica, na qual cada atributo real é representado por uma sequência de bits crescentes. Na configuração adotada, são utilizados 10 bits por atributo, resultando em um vetor binário de  $576 \times 10 = 5.760$  bits por imagem, conforme exemplo na figura 4.2.

Como os quatro modelos de regressão foram definidos de forma independente, foi adicionado uma codificação do tipo *One-Hot Encoding* de 7 dígitos, representando a classe do navio, conforme figura 4.3. Essa estratégia visa incorporar informações categóricas ao vetor de atributos e, assim, capturar diferenças proporcionais entre os tipos de embarcação. O vetor resultante combina os 4.704 valores provenientes da saída intermediária da CNN ( $96 \times 7 \times 7$ ) com os 7 valores do *One-Hot Encoding*, totalizando 4.711 atributos.

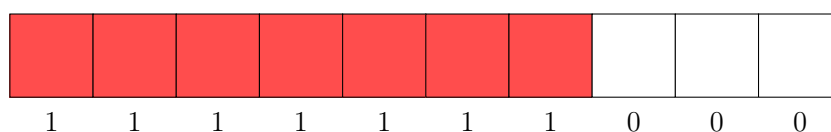


Figura 4.2: Na codificação de termômetro, os bits são ativados progressivamente do menos significativo para o mais significativo, de forma análoga à subida do mercúrio em um termômetro. Na figura, o valor "0.75" é codificado em um termômetro 0 à 1 de resolução 10.

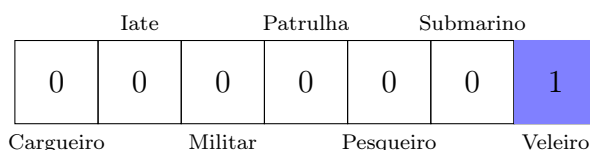


Figura 4.3: Exemplo de vetor One-Hot para sete classes de embarcações. Apenas a posição correspondente à classe é ativada, enquanto todas as demais permanecem inativas. Na figura, a classe "Veleiro" é selecionada.

### 4.3.4 Estrutura dos rótulos

O conjunto de rótulos utilizados para o treinamento e avaliação dos modelos é composto por dois blocos distintos:

- **Rótulo de classificação:** indica o tipo de embarcação presente na imagem, entre 7 possíveis classes, numeradas de 0 a 6. Estes rótulos são codificados como inteiros e utilizados pela instância WiSARD de classificação e pelo modelo YOLOv8 para fins comparativos;
- **Rótulos de regressão:** representam a *bounding box* associada à embarcação detectada. São definidos por quatro valores normalizados no intervalo  $[0, 1]$ , correspondentes às coordenadas do centro da caixa  $(x, y)$  e às suas dimensões  $(w, h)$ , respectivamente, todas em relação ao tamanho da imagem.

Esses dois conjuntos de rótulos são utilizados em paralelo: os valores discretos de classe são tratados pela rede WiSARD, enquanto os valores contínuos das *bounding boxes* são destinados às regressões realizadas pelos modelos responsáveis.

## 4.4 Pipeline Proposto

O sistema proposto para detecção de embarcações é concebido como um *pipeline* modular e eficiente, projetado para atender às restrições típicas de ambientes embarcados, como baixo consumo de energia, memória limitada e ausência de aceleração por hardware. A arquitetura geral combina etapas de pré-processamento e extração de atributos com a aplicação de Redes Neurais Sem Pesos (WiSARD) para a tarefa de classificação e de diferentes modelos de regressão — SVR, kNN e ReWiSARD — para a estimativa das *bounding boxes*

---

**Algorithm 1** Técnica de detecção de navios usando SOD-Edge

---

```
1: Entrada: Dataset de imagem de navios  $D$ 
2: Saída: Modelo WiSARD  $C$  e modelos de regressão  $X$ ,  $Y$ ,  $W$  e  $H$  treinados
3:  $clsData, clsLabels \leftarrow 0$ 
4:  $regData, xRegLabels, yRegLabels, wRegLabels, hRegLabels \leftarrow 0$ 
5:
6: # Pipeline de Preparação
7: for cada imagem  $img$  em  $Dataset$  do
8:   # Reescalamento das Imagens
9:    $img \leftarrow res(img)$  ①
10:
11:   # Extração de Features em 2 pontos
12:    $clsData \leftarrow extracaoFeatures\_1(img)$  ②
13:    $regData \leftarrow extracaoFeatures\_2(img)$  ②
14:
15:   # Normalização e codificação
16:    $clsData \leftarrow codificacaoTermometrica(clsData)$  ③
17:    $regData \leftarrow aplicarOneHotClasse(regData)$  ③
18:
19:   # Extraíndo Ground Truth das anotações pro código
20:    $clsLabels \leftarrow getClsGT(img)$ 
21:    $xRegLabels \leftarrow getXRegGT(img)$ 
22:    $yRegLabels \leftarrow getYRegGT(img)$ 
23:    $wRegLabels \leftarrow getWRegGT(img)$ 
24:    $hRegLabels \leftarrow getHRegGT(img)$ 
25: end for
26:
27: # Pipeline de Detecção
28: # Modelo WiSARD de Classificação
29:  $C = WiSARD.train(clsData, clsLabels)$  ④
30:
31: # Modelos SVR de Regressão
32:  $X = Reg.train(regData, xRegLabels)$  ⑤
33:  $Y = Reg.train(regData, yRegLabels)$  ⑤
34:  $W = Reg.train(regData, wRegLabels)$  ⑤
35:  $H = Reg.train(regData, hRegLabels)$  ⑤
36:
37: return Modelos  $C$ ,  $X$ ,  $Y$ ,  $W$ ,  $H$ 
```

---

### 4.4.1 Pipeline de Preparação

O módulo de preparação tem como objetivo converter a imagem bruta em uma representação numérica compacta e informativa, adequada para as etapas de classificação e regressão subsequentes. Nessa fase, o algoritmo executa transformações sucessivas que reduzem a dimensionalidade dos dados visuais, realçam padrões relevantes e estruturam os atributos no formato esperado pelas redes sem pesos e pelos regressores utilizados. As principais etapas envolvem o pré-processamento das imagens, a extração de atributos por meio de uma CNN leve e a codificação binária compatível com o modelo WiSARD.

1. **Pré-processamento**①: imagens RGB são carregadas e redimensionadas para  $224 \times 224$  pixels;
2. **Extração de atributos**②: aplicação de uma CNN leve e pré-treinada, adequada para dispositivos embarcados, para obtenção de vetores de atributos contínuos em diferentes níveis de abstração;
3. **Normalização e codificação**③: os atributos contínuos são normalizados para o intervalo  $[0, 1]$  e convertidos em vetores binários por meio de codificação termométrica, garantindo compatibilidade com a rede WiSARD;

### 4.4.2 Pipeline de Detecção

O módulo de detecção é responsável por interpretar as representações geradas na etapa anterior e produzir as saídas finais do sistema: a categoria da embarcação e sua localização na imagem. Nessa etapa integram-se dois componentes distintos — classificação e regressão — organizados de modo a explorar diferentes aspectos dos vetores de atributos. Enquanto a rede WiSARD executa a tarefa de identificação do tipo de embarcação, o conjunto de regressões estima as coordenadas e dimensões da caixa delimitadora associada ao objeto detectado.

1. **Classificação**④: o vetor binário é submetido à WiSARD clássica, que realiza a predição do tipo de embarcação entre as categorias consideradas;
2. **Regressão**⑤: em paralelo, um segundo vetor de atributos é direcionado a quatro instâncias independentes de regressores, responsáveis por prever as coordenadas  $(x, y)$  e as dimensões  $(w, h)$  da *bounding box* que delimita a embarcação.

Essa divisão clara entre os módulos de classificação e regressão não apenas favorece a flexibilidade experimental, como também permite a análise isolada do desempenho de cada componente.

#### 4.4.2.1 WiSARD para classificação

A WiSARD (Wilkie, Stonham and Aleksander's *Recognition Device*) é uma rede neural sem pesos baseada em memórias RAM binárias. Cada classe é representada por um conjunto de RAMs que memorizam os padrões binários dos dados de treino. Durante a inferência, cada RAM recebe uma subcadeia do vetor binário de entrada e indica se o padrão foi previamente observado. A classe cuja RAM acumula maior número de acertos é selecionada como predição.

A distribuição das entradas entre as RAMs é tradicionalmente realizada de forma pseudoaleatória, garantindo que diferentes combinações de bits sejam mapeadas para distintas posições de memória. Essa estratégia busca evitar que a correlação direta entre posições vizinhas dos vetores binários introduza redundância no aprendizado, favorecendo uma generalização mais equilibrada.

Entretanto, quando o vetor de entrada representa uma imagem ou um mapa de características convolucional, essa aleatorização pode eliminar correlações espaciais relevantes, uma vez que bits adjacentes podem corresponder a regiões visualmente próximas e semanticamente relacionadas. Nesse sentido, um mapeamento espacialmente em que regiões próximas da imagem sejam associadas a RAMs próximas poderia, em teoria, preservar melhor a estrutura local e beneficiar tarefas de visão computacional.

Neste trabalho optou-se por manter a estratégia pseudoaleatória de distribuição das entradas, preservando o comportamento clássico do modelo original. Essa decisão foi motivada tanto pela necessidade de garantir comparabilidade com implementações canônicas da literatura quanto pelo potencial da aleatorização em conferir robustez frente a ruídos, variações de iluminação e reflexos, características típicas das imagens de ambientes marítimos. Essa discussão abre espaço para investigações futuras acerca do impacto do mapeamento espacial na eficiência e precisão da WiSARD em contextos visuais.

Essa arquitetura apresenta diversas vantagens em relação aos modelos baseados em gradientes:

- Treinamento extremamente rápido, com apenas uma passagem pelos dados;
- Baixo uso de memória e processamento, devido às operações puramente lógicas;
- Possibilidade de aprendizado incremental sem perda de desempenho anterior.

Essas características tornam a WiSARD especialmente adequada para o módulo de classificação deste trabalho, que busca combinar eficiência com interpretabilidade.

#### 4.4.2.2 Modelos de regressão para estimativa de coordenadas

Para a tarefa de regressão das *bounding boxes* — isto é, a predição das coordenadas  $(x, y)$  e das dimensões  $(w, h)$  — são considerados diferentes modelos de regressão, explorando abordagens com distintos níveis de complexidade e pressupostos teóricos. A proposta contempla o uso de *Support Vector Regression* (SVR), *k-Nearest Neighbors* (kNN) e de uma extensão da arquitetura WiSARD adaptada para regressão, denominada *Regression WiSARD* (ReWiSARD).

O SVR, uma extensão das Máquinas de Vetores de Suporte (SVM) para variáveis contínuas, destaca-se pela robustez em espaços de alta dimensionalidade e pela flexibilidade no uso de diferentes funções núcleo (*kernels*). O kNN, por sua vez, caracteriza-se como um método não paramétrico e baseado em instâncias, apresentando simplicidade computacional e comportamento competitivo em cenários com conjuntos de dados reduzidos. Já a ReWiSARD preserva os princípios fundamentais das Redes Neurais Sem Peso, realizando a estimativa de valores contínuos a partir de estruturas de memória associativa, mantendo o aprendizado direto e não iterativo.

Embora técnicas de regressão já tenham sido exploradas em tarefas de visão computacional, como estimativa de pose e análise de atributos visuais, sua aplicação direta à regressão de caixas delimitadoras em sistemas de detecção leves voltados à borda permanece pouco explorada. Neste trabalho, independentemente do modelo adotado, cada regressor é dedicado a um parâmetro específico  $(x, y, w, h)$ , recebendo como entrada os atributos extraídos pela CNN, complementados por um vetor *one-hot* da classe predita pela WiSARD, de forma a incorporar informações contextuais na estimação espacial.

#### 4.4.3 Entrada e saída do modelo

O sistema proposto recebe como entrada imagens RGB de resolução  $224 \times 224$  pixels. Após o processamento, cada imagem é convertida em dois vetores de atributos: um de 576 dimensões, que posteriormente são codificados em 5.760 bits por meio de codificação termométrica para a subtarefa de classificação, e o segundo de 4.711 dimensões, sendo 4.704 da saída da CNN e mais 7 por meio da codificação *One-Hot* da classe da embarcação.

A saída do modelo é composta por uma classificação da embarcação estimada pela WiSARD, entre as sete categorias disponíveis, e as coordenadas  $(x, y)$  e dimensões  $(w, h)$  da sua caixa delimitadora, estimadas por quatro modelos regressores independentes.

#### 4.4.3.1 Considerações adicionais

A arquitetura modular do sistema proposto permite a substituição, reconfiguração e avaliação isolada de cada componente, sem a necessidade de reescrever o *pipeline* completo. Essa característica favorece tanto a experimentação em ambiente computacional quanto as adaptações em ambiente embarcado.

Por exemplo, é possível trocar a CNN utilizada na extração de atributos, testar diferentes esquemas de codificação binária (como termométrica ou *One-Hot*) ou ainda empregar funções alternativas aos modelos de regressão utilizados. Essa flexibilidade é especialmente vantajosa na fase embarcada, em que ajustes finos podem ser necessários para maximizar a eficiência do sistema diante das restrições específicas de hardware.

# Capítulo 5

## Experimentos e Resultados

Este capítulo apresenta os experimentos realizados para avaliar a solução proposta. Primeiramente, descreve-se a implementação, destacando as ferramentas utilizadas, as etapas do desenvolvimento e suas limitações. Em seguida, são apresentados o cenário de experimentos e o conjunto de dados empregados, juntamente com as métricas definidas para análise. Por fim, discutem-se os resultados obtidos nos experimentos conduzidos, com foco na avaliação de desempenho e viabilidade da solução.

### 5.1 Implementação

A implementação do sistema proposto envolveu tanto o desenvolvimento em ambiente computacional quanto em ambiente embarcado. Descrevemos em detalhes técnicos os componentes, bibliotecas, versões utilizadas, configurações específicas e limitações identificadas durante o desenvolvimento.

#### 5.1.1 Recursos utilizados

Para o desenvolvimento do sistema, foram empregados diferentes recursos de hardware e software, conforme listado a seguir:

- **Ambiente Computacional:** Google Colab Pro, com acesso a uma GPU NVIDIA Tesla T4 (15 GB VRAM) e 12.7 GB de RAM disponível. Este ambiente foi utilizado para o primeiro cenário de experimentos, treinamento dos modelos e comparações de desempenho entre a solução proposta e o modelo YOLOv8.
- **Ambiente Embarcado:** Raspberry Pi 4 Model B, equipado com 8 GB de RAM, processador ARM Cortex-A72 quad-core a 1.5 GHz e sistema operacional Raspberry Pi OS 64-bit. Esse dispositivo foi empregado no segundo

cenário dos experimentos para validar a viabilidade de execução do pipeline em uma instância de recursos limitados, simulando aplicações reais na borda.

- **Linguagem de Programação:** Todos os scripts foram desenvolvidos em **Python 3.12.11** (ambiente computacional) e **Python 3.9.2** (ambiente embarcado), garantindo compatibilidade com as bibliotecas descritas a seguir.
- **Bibliotecas:**
  - **wisardpkg** LIMA FILHO *et al.* (2020) — definição e treinamento do modelo WiSARD clássico. Versão utilizada: 1.6.3.
  - **scikit-learn** — implementação do Support Vector Regression (SVR) para a tarefa de regressão das *bounding boxes*. Versão utilizada: 1.6.1.
  - **PyTorch** — utilização da MobileNetV3 Small pré-treinada no ImageNet para extração de *features*. Versão utilizada: 2.8.0 + cu126 e torchvision: 0.23.0 + cu126.
  - **Ultralytics** — implementação e treinamento do modelo YOLOv8 usado como *baseline*. Versão utilizada: 8.3.203.
  - **CodeCarbon** — monitoramento do consumo energético em watt-hora (Wh). Versão utilizada: 3.0.5.
  - **NumPy** — manipulação de vetores de atributos e operações matemáticas. Versão utilizada: 2.0.2.
  - **Matplotlib** — visualização de imagens processadas e resultados de predição. Versão utilizada: 3.10.0.
  - **OpenCV (cv2)** — leitura, transformação e pré-processamento das imagens. Versão utilizada: 4.12.0.88.
  - **Random** — geração de números aleatórios para particionamento de dados e inicialização.
  - **Math** — funções matemáticas auxiliares.
  - **Time** — medições de tempo de pré-processamento, treino e inferência.
  - **OS** — execução de tarefas específicas do sistema operacional.

### 5.1.2 Como foi implementado

Os algoritmos propostos foram estruturado em cinco etapas principais:

1. **Pré-processamento:** todas as imagens foram redimensionadas para  $640 \times 640$  pixels. Para o pipeline WiSARD, houve ainda um reescalamento para  $224 \times 224$ , de modo a adequar a entrada ao extrator de características.

2. **Extração de Características:** foi utilizada a *MobileNetV3 Small*, pré-treinada no ImageNet, como extrator de *features*. Dois pontos da rede foram explorados:

- Saída de `features.9`:  $576 \times 1 \times 1 = 576$  atributos reais por imagem.
- Saída de `features.12`:  $96 \times 7 \times 7 = 4.704$  atributos reais por imagem.

Esses vetores de alta dimensionalidade capturam informações semânticas em diferentes níveis de abstração.

3. **Transformação dos atributos:**

- **Para a classificação:** aplicou-se **binarização**, necessária pois a arquitetura WiSARD exige entradas binárias:
  - **Normalização:** escalonamento dos valores de atributos entre os percentis 5 e 95 da distribuição, com truncamento dos valores fora desse intervalo.
  - **Codificação Termômetro:** cada valor contínuo normalizado foi convertido em um vetor binário de 10 posições, resultando em 5.760 bits por imagem no ponto `features.9`.
- **Para a regressão:** os atributos permaneceram contínuos, e foi adicionado um *one-hot encoding* da classe de cada amostra, de modo a auxiliar os modelos a capturar as proporções específicas de cada tipo de embarcação.

4. **Treinamento:**

- Um modelo WiSARD foi treinado para a sub tarefa de classificação multiclasse das embarcações. Configuração: 12 RAMs de 480 bits cada.
- Quatro instâncias regressoras independentes foram treinadas, cada uma responsável por prever um parâmetro da *bounding box* ( $x, y, w, h$ ). Configuração:
  - SVR:  $\epsilon = 0.02$  e  $C = 0.1$
  - kNN:  $\epsilon = 0.02$  e  $C = 0.1$
  - ReWisard:  $\epsilon = 0.02$  e  $C = 0.1$

5. **Benchmark:** para fins comparativos, foi treinado um modelo YOLOv8 no mesmo conjunto de dados, utilizando pesos pré-treinados como ponto de partida.

### 5.1.3 Limitações

Durante a implementação, foram identificadas algumas limitações importantes:

- **Ausência de multidetecção:** o pipeline atual é capaz de detectar apenas uma embarcação por imagem, não contemplando cenários em que múltiplos objetos estão presentes simultaneamente.

## 5.2 Cenários de Experimentação e Dados

Nesta seção descrevemos os cenários de experimentação utilizados para validar o sistema proposto, bem como as características do conjunto de dados adotado. Dois cenários complementares foram considerados: **computacional** (Google Colab Pro, com GPU dedicada) e **embarcado** (Raspberry Pi 4). O objetivo dessa dualidade é avaliar tanto o desempenho em um ambiente de alto poder computacional quanto a viabilidade prática em um ambiente de recursos restritos, mais próximo de aplicações reais de sistemas embarcados.

### 5.2.1 Ambientes de execução

Dois ambientes distintos foram utilizados para condução dos experimentos:

#### Ambiente Computacional

No ambiente computacional, todos os experimentos foram conduzidos no Google Colab Pro, utilizando uma GPU NVIDIA Tesla T4 com 25 GB de RAM de sistema. Esse ambiente foi escolhido devido à sua disponibilidade, reprodutibilidade e suporte nativo a bibliotecas modernas de aprendizado de máquina.

O uso do Colab Pro permitiu explorar configurações de treinamento com múltiplas épocas e realizar comparações diretas com o estado da arte (YOLOv8). Outro ponto relevante é a facilidade de integração com repositórios no GitHub e com a plataforma Roboflow, utilizada para organizar e anotar os dados.

Esse cenário representa um ambiente de prototipagem acelerada, onde o objetivo principal foi verificar a eficácia do pipeline proposto em condições ideais de processamento, servindo de linha de base para comparação com o ambiente embarcado.

#### Ambiente Embarcado

O segundo ambiente experimental foi configurado em um Raspberry Pi 4 Model B com 8 GB de RAM, processador ARM Cortex-A72 quad-core a 1.5 GHz e sistema operacional Raspberry Pi OS (64 bits).

Nesse cenário, o foco foi validar a viabilidade do pipeline em borda, analisando aspectos de tempo de execução e consumo energético em um dispositivo de baixo custo e amplamente utilizado em aplicações de IoT e visão computacional embarcada.

## 5.2.2 Base de dados

O conjunto de dados utilizado nesta dissertação é composto por 697 imagens RGB, obtidas por meio da plataforma Roboflow, contendo anotações manuais de embarcações em diversos contextos visuais. As imagens apresentam diversidade quanto ao tipo de embarcação e ângulo de visão.

As imagens foram anotadas com caixas delimitadoras (*bounding boxes*) e classificadas em 7 categorias distintas de embarcações. A Tabela 5.1 apresenta a distribuição das classes no conjunto completo:

Tabela 5.1: Distribuição de classes no conjunto de dados

Classe	Número de Amostras
Veleiro	100
Patrulha	99
Pesqueiro	100
Militar	102
Iate	100
Navio mercante	97
Submarino	99

Para o treinamento, validação e avaliação dos modelos, o conjunto de dados foi dividido da seguinte forma:

- **Treinamento (85%)**: 592 imagens;
- **Validação (5%)**: 35 imagens;
- **Teste (10%)**: 70 imagens.

Essa divisão justifica-se pelo tamanho relativamente limitado do conjunto de dados, tornando necessário destinar a maior parcela das amostras ao treinamento, de modo a favorecer a capacidade de generalização dos modelos avaliados. A inclusão de um conjunto de validação, ainda que reduzido, permite a realização de ajustes intermediários, em especial a seleção de hiperparâmetros, sem comprometer a independência do conjunto de teste. Por fim, a alocação de 10% das imagens para teste assegura a existência de um conjunto independente e representativo para a avaliação final do desempenho.

O conjunto de validação foi utilizado especificamente para a seleção dos hiperparâmetros dos modelos baseados em WiSARD e regressão (SVR, kNN e ReWiSARD). Para cada configuração avaliada, os modelos foram treinados exclusivamente com o conjunto de treinamento e avaliados no conjunto de validação, considerando a acurácia como critério de comparação. As configurações que apresentaram melhor desempenho no conjunto de validação foram então selecionadas e mantidas fixas nos experimentos subsequentes.

A separação dos dados foi realizada com estratificação por classe e uso de uma semente pseudoaleatória fixa, garantindo a reprodutibilidade dos experimentos. As imagens do conjunto de teste foram utilizadas exclusivamente para a avaliação final, não sendo empregadas em nenhuma etapa de ajuste de parâmetros ou tomada de decisão, assegurando rigor na comparação de desempenho entre as diferentes abordagens.

### 5.2.3 Justificativa da Escolha

O dataset adotado reflete cenários reais de navegação em mar aberto, contemplando variações de iluminação, perspectiva e tipos de embarcação. Além disso, seu formato de anotação em YOLOv8 facilitou tanto a adaptação ao modelo de referência quanto a conversão necessária para alimentar o pipeline WiSARD.

Outro fator determinante para a escolha foi o fato de que as imagens possuem apenas uma instância de embarcação por cena, o que está alinhado com a limitação atual do modelo desenvolvido, capaz de realizar apenas uma detecção por imagem.

Por essas características, o conjunto foi considerado adequado para a investigação da hipótese central desta dissertação: avaliar a viabilidade do uso de redes neurais sem pesos para detecção embarcada de embarcações.

## 5.3 Métricas

A avaliação do sistema proposto foi conduzida por meio de um conjunto de métricas que cobrem quatro dimensões principais: desempenho em classificação, precisão em regressão, eficiência temporal e eficiência energética. Esta escolha buscou garantir uma análise abrangente, considerando tanto a qualidade das predições quanto a viabilidade de implementação em cenários computacionais restritos, como os dispositivos embarcados.

### 5.3.1 Métricas de Classificação

Para avaliar o desempenho da rede WiSARD clássica na tarefa de classificação multiclasse das embarcações, foram adotadas as seguintes métricas:

- **Precisão (*Precision*):**

$$Precision = \frac{TP}{TP + FP}$$

Mede a proporção de exemplos classificados corretamente como positivos em relação ao total de exemplos classificados como positivos. Essa métrica é especialmente relevante no contexto de detecção de embarcações, pois reduz o risco de falsos alarmes (falsos positivos), que em aplicações de monitoramento poderiam gerar alertas indevidos e comprometer a confiabilidade do sistema.

- **Revocação (*Recall*):**

$$Recall = \frac{TP}{TP + FN}$$

Mede a capacidade do modelo em identificar todos os exemplos positivos, ou seja, a proporção de verdadeiros positivos em relação ao total de positivos existentes. Crítica no contexto marítimo, pois valores baixos de revocação indicam que embarcações reais deixaram de ser detectadas (falsos negativos), comprometendo diretamente a segurança e a eficácia do sistema.

- **F1-Score:**

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Representa a média harmônica entre precisão e revocação, equilibrando os dois aspectos. Essa métrica é particularmente útil em cenários onde pode haver flutuações entre as classes, garantindo uma visão mais balanceada do desempenho geral do modelo.

### 5.3.2 Métricas de Regressão

Para a tarefa de regressão, referente à predição dos parâmetros da *bounding box* (coordenadas  $(x, y)$  e dimensões  $(w, h)$ ), foram utilizadas duas métricas complementares:

- **Erro Absoluto Médio (*Mean Absolute Error* - MAE):**

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

onde  $y_i$  é o valor real e  $\hat{y}_i$  é a predição.

O MAE foi escolhido por sua interpretação direta e pelo fato de não penalizar desproporcionalmente erros maiores. Além disso, no presente trabalho, a regressão foi realizada com quatro modelos SVR independentes — um para

cada parâmetro da caixa delimitadora. Dessa forma, o MAE permite avaliar o erro de cada parâmetro separadamente, evidenciando quais dimensões apresentam maior dificuldade de ajuste. Essa granularidade é importante para compreender limitações estruturais do modelo.

- ***Intersection over Union (IoU)***:

$$IoU = \frac{rea(B_{pred} \cap B_{real})}{rea(B_{pred} \cup B_{real})}$$

onde  $B_{pred}$  representa a caixa predita e  $B_{real}$  a caixa anotada.

Enquanto o MAE mede os erros individuais de cada parâmetro, o IoU oferece uma visão integrada da qualidade da regressão, avaliando se a combinação dos quatro parâmetros resulta em uma caixa coerente e bem sobreposta ao objeto real. Assim, o uso conjunto de MAE e IoU possibilita tanto a análise detalhada quanto a validação global da predição das caixas delimitadoras.

### 5.3.3 Métricas de Eficiência Temporal

Considerando a relevância da aplicação em sistemas de tempo real e embarcados, foram utilizadas métricas de tempo para cada etapa do pipeline:

- **Tempo de Pré-processamento**: intervalo necessário para redimensionar, normalizar, extrair características e binarizar as imagens.
- **Tempo de Treinamento**: tempo total gasto para o aprendizado do modelo. Essa métrica evidencia a diferença entre abordagens leves, como a WiSARD (treino quase instantâneo), e modelos mais pesados, como o YOLOv8, que dependem de retropropagação intensiva.
- **Tempo de Inferência**: tempo requerido para gerar predições em novas imagens. É crítica em cenários de monitoramento contínuo pois a capacidade de resposta em tempo real pode definir a viabilidade prática da solução.

Essas três métricas foram medidas separadamente para evidenciar gargalos específicos e permitir comparações detalhadas entre a proposta WiSARD-SVR e YOLOv8, tanto no ambiente computacional quanto no embarcado.

### 5.3.4 Métricas de Eficiência Energética

Por fim, foi avaliada a eficiência energética dos modelos, fundamental em cenários embarcados com recursos limitados de energia. Para isso, utilizou-se a seguinte métrica:

- **Consumo Energético (Wh):**

$$E = P \times t$$

onde  $P$  representa a potência média consumida e  $t$  o tempo de execução.

O consumo energético foi monitorado com a ferramenta CodeCarbon, que estima o gasto em watt-hora (Wh) em cada etapa do pipeline (pré-processamento, treino e inferência). Essa métrica é essencial para comparar o custo energético relativo de cada abordagem e avaliar sua viabilidade em sistemas embarcados, onde a autonomia energética pode ser tão determinante quanto a acurácia do modelo.

### 5.3.5 Resumo das Métricas

A tabela 5.2 resume as métricas adotadas em cada experimento e sua finalidade.

Dimensão	Métrica	Objetivo
Classificação	Precisão	Reduzir falsos positivos, garantindo confiabilidade
	Revocação	Reduzir falsos negativos, evitando omissão de embarcações
	F1-Score	Equilibrar precisão e revocação em bases desbalanceadas
Regressão	MAE	Avaliar erro médio individual em $(x, y, w, h)$
	IoU	Avaliar a qualidade global da sobreposição das caixas
Eficiência Temporal	Pré-processamento	Medir custo de preparação dos dados
	Treinamento	Comparar custo de aprendizado dos modelos
Eficiência Energética	Inferência	Avaliar viabilidade em tempo real
	Consumo (Wh)	Analisar custo energético por etapa

Tabela 5.2: Resumo das métricas utilizadas nos experimentos e seus objetivos.

## 5.4 Experimentos

A instância experimental definida para validação do sistema proposto compreende a forma como os experimentos foram estruturados. O objetivo é garantir a comparabilidade entre abordagens distintas e a representatividade de cenários reais, tanto em ambiente com suporte computacional robusto quanto em dispositivos embarcados de baixo custo.

Foram conduzidos quatro experimentos principais, cada um voltado para avaliar um aspecto distinto do desempenho dos modelos. Os experimentos 3 e 4 foram realizados tanto no ambiente computacional quanto no ambiente embarcado, respeitando as limitações de cada abordagem: o modelo YOLOv8 foi treinado e avaliado exclusivamente no ambiente computacional enquanto o pipeline proposto foi integralmente executado em ambos os ambientes.

#### 5.4.1 Experimento 1: Classificação

**Objetivo:** Avaliar a capacidade dos modelos em identificar corretamente a classe da embarcação nas imagens, comparando a rede WiSARD clássica ao modelo YOLOv8.

**Métricas:** Precisão, Revocação e F1-Score.

#### 5.4.2 Experimento 2: Regressão das caixas delimitadoras

**Objetivo:** Medir a acurácia na predição das caixas delimitadoras a partir de modelos SVR, comparados ao YOLOv8.

**Métricas:** MAE, calculado separadamente para cada parâmetro  $(x, y, w, h)$  a fim de detalhar os erros em posição e escala e IoU, que avalia a sobreposição entre caixas previstas e anotadas, fornecendo uma visão global da qualidade da detecção.

#### 5.4.3 Experimento 3: Eficiência Temporal

**Objetivo:** Avaliar o custo computacional do pipeline em cada etapa — pré-processamento, treinamento e inferência — considerando tanto o ambiente computacional quanto o embarcado.

**Métricas:** Tempo de Pré-processamento (incluindo redimensionamento, normalização, extração de características e binarização), Tempo de Treinamento (atualizações de RAM no caso da WiSARD e backpropagation no YOLOv8) e Tempo de Inferência (produção de uma predição em nova amostra, relevante em cenários de tempo real).

#### 5.4.4 Experimento 4: Eficiência Energética

**Objetivo:** Quantificar o consumo energético associado ao uso dos modelos, analisando as principais etapas do pipeline em diferentes cenários de execução.

**Métrica:** Consumo energético em Wh, estimado com a biblioteca `codecarbon`, contabilizando CPU e GPU. Essa métrica é especialmente importante para sistemas embarcados com autonomia energética limitada.

## Resumo da execução

Os quatro experimentos — classificação, regressão, eficiência temporal e eficiência energética — foram aplicados de forma sistemática a cada abordagem, respeitando as características e restrições de cada ambiente. Essa estrutura garante que os resultados obtidos sejam comparáveis e representativos para o problema de detecção de embarcações em cenários reais.

## 5.5 Avaliação de Resultados

Nesta seção são discutidos os resultados obtidos nos quatro experimentos descritos anteriormente, considerando tanto o ambiente computacional (Google Colab Pro, com GPU dedicada) quanto o ambiente embarcado (Raspberry Pi 4). Como os experimentos de classificação e regressão dependem apenas dos dados e do algoritmo, seus resultados são equivalentes nos dois cenários. As diferenças relevantes entre os ambientes concentram-se, portanto, nos experimentos de eficiência temporal e energética, diretamente impactados pelas limitações de hardware do Raspberry Pi 4.

### 5.5.1 1º Experimento – Classificação

Os resultados de classificação indicam um comportamento distinto entre os modelos avaliados. O YOLOv8 apresentou maior revocação, identificando a maior parte das embarcações presentes nas imagens. A rede WiSARD, por outro lado, obteve maior precisão, reduzindo a ocorrência de falsos positivos.

O F1-Score evidencia que ambas as abordagens apresentam desempenho competitivo, cada uma privilegiando aspectos diferentes da tarefa: o YOLOv8 prioriza abrangência na detecção, enquanto a WiSARD favorece a confiabilidade das predições. Observa-se ainda que a WiSARD atinge tais resultados com uma arquitetura substancialmente mais leve, característica de relevância para aplicações em cenários com recursos computacionais restritos.

Métrica	Proposta WiSARD	YOLOv8
Precisão	$0.903 \pm 0.021$	0.883
Revocação	$0.891 \pm 0.024$	0.976
F1-Score	$0.892 \pm 0.024$	0.927

Tabela 5.3: Resultados médios de classificação com WiSARD e YOLO

**Matriz de Confusão - Classificação WiSARD**

Ground Truth \ Predição	Pesqueiro	Mercante	Militar	Patrulha	Veleiro	Submarino	late
Pesqueiro	0.75	0.12	0.00	0.12	0.00	0.00	0.00
Mercante	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Militar	0.00	0.00	0.78	0.00	0.00	0.00	0.22
Patrulha	0.00	0.00	0.11	0.89	0.00	0.00	0.00
Veleiro	0.06	0.00	0.00	0.06	0.89	0.00	0.00
Submarino	0.00	0.00	0.00	0.00	0.00	1.00	0.00
late	0.00	0.00	0.11	0.00	0.00	0.00	0.89

Figura 5.1: Matriz de confusão dos resultados de classificação com a rede WiSARD, onde as linhas representam as classes reais e as colunas indicam as predições. As células diagonais evidenciam as instâncias corretamente classificadas, enquanto os valores fora da diagonal revelam confusões entre classes visualmente semelhantes, navios militares e patrulha.

### 5.5.2 2º Experimento – Regressão

Na tarefa de regressão das *bounding boxes*, foram avaliadas três abordagens: SVR, kNN e Regression WiSARD (ReWiSARD). Os resultados foram comparados a um modelo profundo de referência, o YOLOv8.

Os resultados apontaram o SVR como a abordagem mais consistente entre os modelos avaliados. O método apresentou desempenho competitivo na predição das coordenadas de posição  $(x, y)$ , aproximando-se dos resultados obtidos pelo YOLOv8, e superou tanto o kNN quanto o ReWiSARD nos parâmetros de escala  $(w, h)$ , evidenciando maior capacidade de generalização geométrica entre as abordagens analisadas. Esse comportamento reforça a adequação do SVR para tarefas de localização, especialmente quando operando sobre vetores de características compactos extraídos por redes convolucionais pré-treinadas.

O kNN, embora tenha apresentado erros reduzidos na predição das coordenadas centrais, com valores comparáveis aos do SVR para  $(x, y)$ , mostrou limitações mais evidentes na estimativa dos parâmetros de escala, refletindo-se em erros médios superiores em  $(w, h)$  e em um valor médio de IoU inferior ao alcançado pelo SVR. Ainda assim, o método destacou-se pelo custo computacional extremamente baixo na fase de treinamento, o que o torna atrativo em cenários onde simplicidade e rapidez de ajuste são requisitos centrais.

A abordagem baseada em ReWiSARD apresentou o desempenho mais modesto entre os modelos avaliados, com erros médios absolutos mais elevados e menor valor de IoU. Apesar disso, sua inclusão no experimento é relevante do ponto de vista exploratório, uma vez que investiga a extensão do paradigma de redes neurais sem pesos para tarefas de regressão contínua, um problema ainda pouco consolidado na literatura. Os resultados indicam que, embora conceitualmente eficiente para classificação, a ReWiSARD enfrenta limitações estruturais na modelagem precisa de variações contínuas de escala e proporção.

A análise dos tempos de processamento reforça as diferenças entre as abordagens. O SVR manteve um equilíbrio favorável entre custo de treinamento e inferência, enquanto o kNN apresentou treinamento praticamente instantâneo, à custa de maior dependência do conjunto de dados na fase de inferência. Por sua vez, a ReWiSARD demandou maior tempo de treinamento, decorrente da construção de suas estruturas de memória distribuída, e apresentou o maior custo de inferência entre os modelos leves analisados.

Quando comparados ao YOLOv8, que alcançou IoU médio de 0.912, observa-se que nenhum dos modelos leves atingiu o mesmo nível de precisão geométrica. Ainda assim, os resultados obtidos pelo SVR demonstram que é possível aproximar-se do desempenho de arquiteturas profundas consolidadas utilizando modelos substancialmente mais simples, com menor complexidade estrutural e maior potencial de eficiência energética.

Em síntese, os resultados posicionam o SVR como a melhor alternativa leve para a tarefa de regressão dentro da proposta híbrida, enquanto kNN e ReWiSARD cumprem papéis complementares, fornecendo evidências experimentais sobre os limites e possibilidades de métodos baseados em distância e redes neurais sem pesos em cenários de detecção na borda.

Parâmetro	Proposta			YOLO
	SVR	kNN	ReWiSARD	
$x$	0.0241	0.0239	0.0286	0.0101
$y$	0.0305	0.0310	0.0387	0.0134
$w$	0.0703	0.0942	0.1355	0.0250
$h$	0.0704	0.0976	0.1446	0.0327
<b>IoU</b>	0.760	0.715	0.632	0.912

Tabela 5.4: Resultados da regressão dos parâmetros da *bounding box* utilizando SVR, kNN, ReWiSARD e YOLOv8.

### 5.5.3 3<sup>o</sup> Experimento – Eficiência Temporal

**Objetivo:** Medir o custo temporal das principais etapas do pipeline (pré-processamento, treinamento e inferência), com foco na comparação entre as diferentes combinações da WiSARD com algoritmos de regressão, bem como em relação ao YOLOv8.

#### Ambiente Computacional

No ambiente do Google Colab Pro, os resultados de eficiência temporal evidenciam que o pipeline proposto apresenta custo computacional significativamente inferior ao do YOLOv8 em todas as etapas avaliadas. O tempo de pré-processamento, que inclui normalização das imagens e extração de *features* por meio de uma rede convolucional leve, foi de aproximadamente 12.8 segundos. Esse valor é comum a todas as combinações da proposta, uma vez que antecede a etapa de regressão, e contrasta fortemente com o custo de mais de 25 minutos observado para o preparo e treinamento do YOLOv8.

Na etapa de treinamento, a WiSARD apresentou custo extremamente reduzido, concluindo seu aprendizado em aproximadamente 0.23 segundos. O SVR demandou alguns segundos adicionais para o ajuste dos parâmetros regressivos, resultando em um tempo total de treinamento ainda muito inferior ao do modelo profundo. Quando comparadas às alternativas avaliadas, como kNN e ReWiSARD, a combinação WiSARD–SVR mostrou-se mais equilibrada, uma vez que o kNN, embora não possua treinamento explícito, transfere o custo computacional para a fase de inferência, enquanto a ReWiSARD apresentou maior custo de aprendizado devido à construção de suas estruturas de memória distribuída.

Na inferência, a combinação WiSARD–SVR apresentou o menor custo temporal global entre as abordagens avaliadas. O SVR atingiu tempos médios da ordem de milissegundos (0.0029 s), enquanto a WiSARD manteve inferência rápida (0.045 s), resultando em um pipeline altamente eficiente. Em contraste, tanto o kNN quanto a ReWiSARD apresentaram tempos de inferência superiores, ainda que significativamente menores que os do YOLOv8, que demandou aproximadamente 42.8 segundos nessa etapa.

De forma geral, os resultados indicam que a combinação WiSARD–SVR representa o melhor compromisso entre custo temporal e desempenho entre as arquiteturas leves analisadas. O pipeline proposto mantém um custo global de processamento pelo menos duas ordens de grandeza inferior ao do YOLOv8, reforçando sua adequação para aplicações com restrições severas de tempo e recursos computacionais, especialmente em cenários de borda.

Etapa	Proposta				YOLOv8
	WiSARD	SVR	kNN	ReWiSARD	
Pré-processamento			12.776		1539.44
Treinamento	$0.2287 \pm 0.0062$	$4.148 \pm 0.0464$	0.0057	22.3157	–
Inferência	$0.0454 \pm 0.0030$	$0.0029 \pm 0.0001$	$0.1897 \pm 0.0049$	0.3453	42.763

Tabela 5.5: Custo temporal (em segundos) do pipeline proposto utilizando WiSARD combinada com SVR, kNN e ReWiSARD, em comparação ao YOLOv8.

### Ambiente Embarcado

No ambiente embarcado, representado pelo Raspberry Pi 4, a solução proposta apresentou comportamento temporal consistente e alinhado às tendências observadas no ambiente computacional. O pré-processamento destacou-se como a etapa mais custosa em termos de tempo de execução, com valor absoluto elevado quando comparado às demais fases do pipeline. Esse resultado é compatível com o observado no ambiente computacional e reforça que essa etapa depende majoritariamente das operações de manipulação de imagem e extração de características, sendo pouco sensível às diferenças de poder computacional entre as plataformas avaliadas.

No que se refere à etapa de regressão, foram avaliadas três abordagens distintas: SVR, kNN e ReWiSARD. Observou-se que o kNN apresentou tempo de treinamento extremamente reduzido, característica inerente a modelos baseados em instâncias, nos quais o treinamento consiste essencialmente no armazenamento dos exemplos. Em contrapartida, o custo de inferência foi significativamente mais elevado, refletindo a necessidade de cálculos de distância em relação a todo o conjunto de treinamento, o que impacta diretamente o desempenho em dispositivos com recursos limitados.

A abordagem ReWiSARD, por sua vez, apresentou um comportamento inverso, com tempo de treinamento bem mais elevado. Entretanto, embora a inferência também tenha apresentado custo superior ao observado para o SVR, os valores permanecem dentro de uma faixa compatível com aplicações embarcadas, evidenciando um compromisso entre custo de treinamento e capacidade de inferência.

Por fim, a regressão baseada em SVR manteve-se como a alternativa mais equilibrada do ponto de vista temporal, apresentando tempos de inferência reduzidos e treinamento moderado quando comparada às demais abordagens. Em conjunto, os resultados indicam que, mesmo em um dispositivo de baixo custo e recursos computacionais limitados, a solução proposta mantém desempenho temporal competitivo, reforçando sua viabilidade para aplicações embarcadas de detecção e monitoramento de embarcações em tempo quase real.

Etapa	Proposta			
	WiSARD	SVR	kNN	ReWiSARD
Pré-processamento			63.70	
Treinamento	$0.5341 \pm 0.0042$	$9.7474 \pm 0.0667$	$0.0152 \pm 0.0002$	$48.5485 \pm 0.8320$
Inferência	$0.1217 \pm 0.0009$	$0.0107 \pm 0.0001$	$0.3245 \pm 0.0089$	$0.7705 \pm 0.0153$

Tabela 5.6: Custo temporal médio (em segundos), com desvio padrão, da solução proposta no ambiente embarcado (Raspberry Pi 4).

#### 5.5.4 4<sup>o</sup> Experimento – Eficiência Energética

**Objetivo:** Quantificar o consumo energético dos modelos em cada etapa do pipeline, analisando separadamente a tarefa de classificação, realizada pela WiSARD, e a tarefa de regressão, conduzida por diferentes algoritmos.

##### Ambiente Computacional

Os resultados de consumo energético evidenciam a importância de analisar separadamente as contribuições de cada componente do pipeline. A etapa de pré-processamento, responsável pela extração das *features* convolucionais, apresentou consumo energético de aproximadamente 378 mWh e é comum a todas as configurações avaliadas, independentemente do modelo de classificação ou regressão utilizado.

Na tarefa de classificação multiclasse, a WiSARD apresentou consumo energético extremamente reduzido. O treinamento demandou aproximadamente 6.9 mWh, enquanto a inferência consumiu cerca de 1.1 mWh, evidenciando a natureza leve e energeticamente eficiente das redes neurais sem peso. Esses valores confirmam a adequação da WiSARD para cenários nos quais o aprendizado rápido e o baixo consumo energético são requisitos centrais, especialmente em aplicações de computação de borda.

Na tarefa de regressão das *bounding boxes*, foram avaliados três algoritmos distintos: SVR, kNN e ReWiSARD. Os resultados mostram que a escolha do regressor exerce impacto direto no consumo energético do pipeline, mesmo quando operando sobre o mesmo conjunto de *features*.

O SVR apresentou consumo energético moderado durante o treinamento, da ordem de centenas de mWh, associado ao processo de ajuste dos parâmetros do modelo. Em contrapartida, seu custo energético de inferência foi extremamente baixo, inferior a 0.1 mWh, configurando-se como a alternativa mais eficiente para execução contínua e aplicações em tempo real.

O kNN destacou-se pelo consumo energético praticamente desprezível na fase de treinamento, reflexo direto da ausência de um processo explícito de aprendizado.

No entanto, essa vantagem foi acompanhada por um custo de inferência superior ao do SVR, uma vez que o método requer a comparação direta com as amostras armazenadas. Ainda assim, o consumo energético total do kNN permaneceu baixo em termos absolutos, mantendo-se compatível com aplicações de baixa potência.

A ReWiSARD apresentou comportamento distinto em relação às demais abordagens. Embora tenha mantido consumo energético reduzido na inferência, inferior a 1 mWh, seu custo energético de treinamento foi significativamente mais elevado, alcançando valores da ordem de centenas de mWh. Esse resultado reflete o overhead associado à construção das estruturas de memória distribuída da rede, indicando que, no contexto da regressão contínua, a ReWiSARD impõe maior custo energético durante a fase de aprendizado.

Em contraste, o YOLOv8 apresentou consumo energético substancialmente superior em todas as etapas avaliadas, com valores de dezenas de milhares de mWh no pré-processamento e treinamento e centenas de milhares de mWh na inferência. Esses números refletem o custo intrínseco de arquiteturas profundas baseadas em retropropagação e no uso intensivo de GPU.

De forma geral, os resultados indicam que a WiSARD é altamente eficiente energeticamente para a tarefa de classificação, enquanto, entre os regressores avaliados, o SVR apresentou o melhor compromisso entre custo energético e desempenho. O kNN surge como alternativa viável em cenários que priorizam simplicidade e custo mínimo de aprendizado, ao passo que a ReWiSARD se mostra eficiente na inferência, porém energeticamente mais custosa no treinamento. Esses achados reforçam a adequação do pipeline proposto para aplicações com severas restrições energéticas, típicas de ambientes de computação de borda.

Etapa	Proposta				YOLOv8
	WiSARD	SVR	kNN	ReWiSARD	
Pré-processamento					35839.500
Treinamento	6.854	127.548	0.144	595.054	–
Inferência	1.119	0.058	2.313	0.700	862.200

Tabela 5.7: Consumo energético (em mWh) das diferentes etapas do pipeline proposto, comparando a WiSARD (classificação) e os algoritmos de regressão SVR, kNN e ReWiSARD com o YOLOv8.

### Ambiente Embarcado

No ambiente embarcado, o consumo energético seguiu a mesma tendência observada no ambiente computacional, com o pré-processamento representando a parcela mais significativa do gasto total de energia. Esse comportamento reforça que essa etapa

é dominada por operações de manipulação de imagens e extração de características, sendo pouco sensível ao modelo empregado nas etapas subsequentes do pipeline.

No que se refere à etapa de classificação, a rede WiSARD apresentou consumo energético reduzido tanto durante o treinamento quanto na inferência. Esse resultado é coerente com a natureza das redes neurais sem peso, cujo processo de aprendizado baseia-se em operações de escrita em memória, sem etapas iterativas de otimização. Tal característica torna a WiSARD particularmente adequada para execução em dispositivos embarcados, nos quais a eficiência energética é um requisito fundamental.

Em relação à etapa de regressão, foram avaliadas as abordagens SVR, kNN e ReWiSARD, as quais apresentaram perfis energéticos distintos. A regressão baseada em SVR apresentou consumo energético elevado durante o treinamento, associado ao processo de otimização iterativa necessário para o ajuste dos modelos regressivos. Em contrapartida, a fase de inferência mostrou-se altamente eficiente do ponto de vista energético, com consumo reduzido, evidenciando um perfil equilibrado para aplicações embarcadas que priorizam baixo custo energético durante a execução contínua.

O kNN destacou-se pelo consumo energético extremamente reduzido durante o treinamento, característica inerente a modelos baseados em instâncias, nos quais não há fase de otimização propriamente dita. Em contrapartida, o consumo energético durante a inferência foi superior ao observado para o SVR, refletindo o custo associado ao cálculo de distâncias em relação a todo o conjunto de treinamento, o que pode impactar a eficiência em cenários de execução prolongada.

A abordagem ReWiSARD apresentou um consumo energético mais elevado na fase de treinamento, decorrente da construção e escrita das estruturas internas baseadas em redes neurais sem pesos adaptadas para regressão. Entretanto, durante a inferência, o modelo demonstrou elevada eficiência energética, com valores inferiores aos observados para o kNN e comparáveis aos do SVR, evidenciando sua adequação para cenários de execução contínua em dispositivos embarcados.

De forma geral, os resultados indicam que, apesar das diferenças nos perfis energéticos das abordagens avaliadas, a solução proposta mantém consumo energético compatível com dispositivos de baixo custo e recursos computacionais limitados, como o Raspberry Pi 4. Esses achados reforçam a viabilidade da proposta para aplicações embarcadas de monitoramento e detecção de embarcações, especialmente em cenários que demandam operação prolongada e eficiência energética.

Etapa	Classificação WiSARD	Regressão		
		SVR	kNN	ReWiSARD
Pré-processamento		708.426		
Treinamento	6.777	134.633	0.0207	66.135
Inferência	1.537	0.131	0.5962	0.0756

Tabela 5.8: Consumo energético (em mWh) da solução proposta no ambiente embarcado (Raspberry Pi 4).

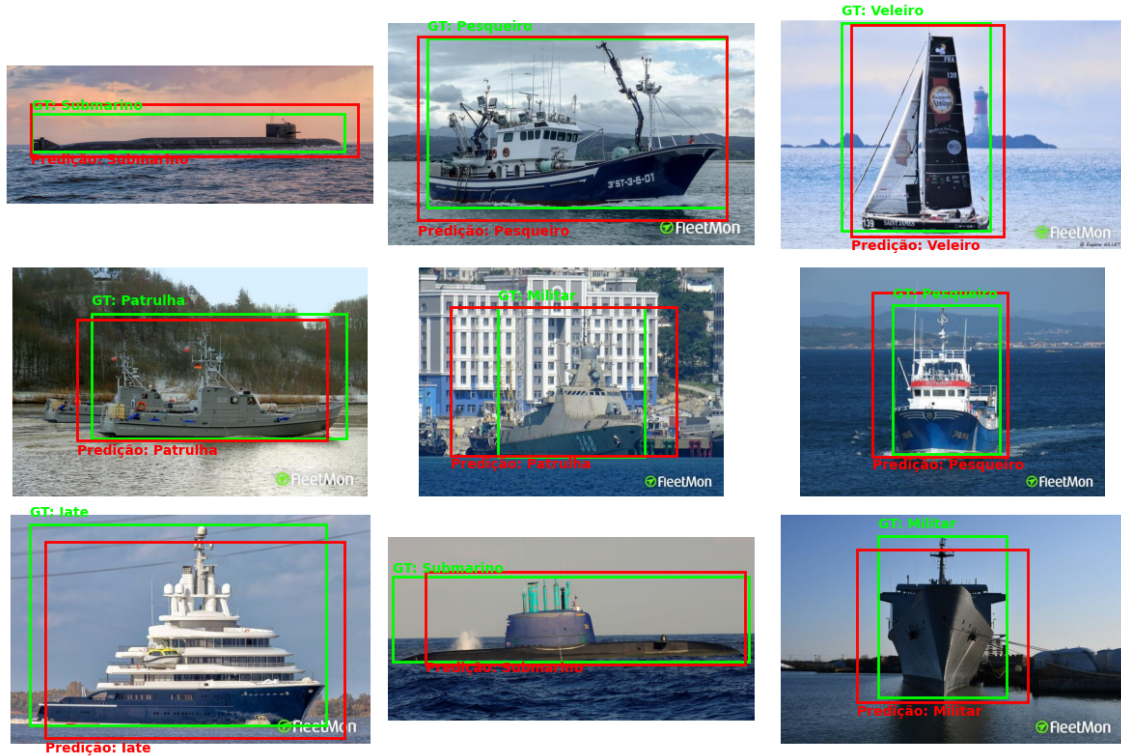


Figura 5.2: Resultados qualitativos da abordagem proposta utilizando o modelo SVR como regressor. As caixas em verde representam as anotações do *ground truth* (GT), enquanto as caixas em vermelho indicam as predições obtidas pelo modelo.

### 5.5.5 Discussão Geral

A análise integrada dos quatro experimentos conduzidos permite sintetizar as principais forças, limitações e compromissos do pipeline proposto, baseado na combinação de redes neurais sem pesos para classificação (WiSARD) e diferentes estratégias de regressão, em contraste com o modelo de referência baseado em aprendizado profundo (YOLOv8).

No experimento de classificação, a WiSARD demonstrou desempenho competitivo em relação ao YOLOv8. Embora o modelo profundo tenha alcançado valores

superiores de revocação, a WiSARD destacou-se em precisão, reduzindo a ocorrência de falsos positivos. O equilíbrio observado no F1-Score confirma que classificadores baseados em redes sem pesos podem alcançar resultados satisfatórios mesmo em tarefas de discriminação multiclasse, com custo computacional significativamente inferior.

No experimento de regressão, o SVR apresentou erros médios próximos aos do YOLOv8 nos parâmetros de localização  $(x, y)$ , evidenciando boa capacidade de estimar a posição central das embarcações. As diferenças mais acentuadas surgiram nos parâmetros de escala  $(w, h)$ , nos quais o YOLOv8 obteve valores inferiores de erro. Ainda assim, a análise baseada no índice de IoU demonstrou que o pipeline proposto é capaz de gerar previsões globais adequadas, alcançando sobreposições consistentes entre as caixas previstas e as anotações de referência.

A inclusão das abordagens kNN e ReWiSARD na etapa de regressão permitiu ampliar a análise, evidenciando que não existe uma única solução ótima, mas sim diferentes compromissos entre custo computacional, consumo energético e desempenho temporal. O kNN destacou-se pelo custo de treinamento praticamente desprezível, à custa de maior tempo e consumo energético durante a inferência, em razão de sua natureza baseada em distância. A ReWiSARD, por sua vez, apresentou maior custo de treinamento, associado à construção das estruturas internas regressivas, mas demonstrou elevada eficiência energética durante a inferência, tornando-se uma alternativa atrativa para cenários de execução contínua em dispositivos embarcados.

Os maiores contrastes tornaram-se evidentes nos experimentos de eficiência temporal e energética. No ambiente computacional, o treinamento da WiSARD foi praticamente instantâneo, enquanto os métodos de regressão apresentaram custos variados conforme sua natureza. Em contrapartida, o YOLOv8 demandou dezenas de minutos apenas para as etapas de pré-processamento e treinamento. Durante a inferência, as combinações baseadas em WiSARD mantiveram tempos compatíveis com aplicações em tempo quase real, reforçando a viabilidade prática do pipeline proposto.

No que se refere ao consumo energético, o pipeline baseado em redes sem pesos apresentou vantagens expressivas. Enquanto o YOLOv8 consumiu dezenas de milhares de mWh durante o treinamento, a solução proposta operou com consumo energético duas ordens de grandeza inferior. Dentro do próprio pipeline, a WiSARD mostrou-se energeticamente irrelevante, com consumo praticamente desprezível. A escolha do método de regressão, entretanto, impactou diretamente o perfil energético do sistema, indicando que aplicações que priorizam baixo consumo durante a inferência podem se beneficiar do uso da ReWiSARD, mesmo à custa de maior tempo de treinamento, enquanto cenários com treinamento frequente e inferência esporádica podem favorecer abordagens como o kNN.

No ambiente embarcado, os resultados confirmaram a robustez e a portabilidade da proposta. Tanto os tempos de execução quanto os consumos energéticos mantiveram-se em patamares coerentes com os observados no ambiente computacional, com variações proporcionais às limitações de hardware. O desempenho obtido no Raspberry Pi 4 evidencia que o pipeline proposto preserva sua eficiência mesmo em condições de recursos restritos, consolidando sua aplicabilidade em cenários reais de monitoramento marítimo.

Apesar desses avanços, é importante destacar as limitações do trabalho. O pipeline atual foi projetado para a detecção de uma única embarcação por imagem, não contemplando cenários de multidetecção. Além disso, em contextos altamente complexos, como ambientes com múltiplos alvos simultâneos ou forte variação de escala, modelos profundos com maior poder de representação, como o YOLOv8, continuam sendo relevantes, ainda que à custa de maior consumo computacional e energético. Nesse sentido, as abordagens avaliadas devem ser vistas como complementares: enquanto o pipeline baseado em redes sem pesos se mostra mais adequado a dispositivos embarcados de baixo custo e baixa autonomia, arquiteturas profundas permanecem valiosas em aplicações de maior porte, nas quais os recursos de processamento e energia não constituem fatores limitantes.

# Capítulo 6

## Conclusão

Este capítulo apresenta o fechamento da dissertação, consolidando as principais contribuições e resultados alcançados ao longo do trabalho. Primeiramente, realiza-se a visão geral das etapas desenvolvidas e dos objetivos atingidos. Em seguida, são discutidas as limitações da pesquisa, que delimitam o escopo da solução proposta e evidenciam oportunidades de melhoria. Na sequência, são indicados possíveis caminhos para trabalhos futuros, capazes de expandir e aprofundar a investigação aqui conduzida. Por fim, é apresentado um arremate final, ressaltando as implicações do estudo e sua relevância no contexto da visão computacional embarcada.

### 6.1 Visão geral do Trabalho

Esta dissertação teve como objetivo investigar a viabilidade de utilizar redes neurais sem pesos em conjunto com diferentes estratégias de regressão para a tarefa de detecção de embarcações em imagens. O foco esteve voltado para cenários embarcados e de recursos restritos, nos quais o consumo energético e o tempo de execução tornam-se fatores críticos para a viabilidade prática das soluções propostas.

O problema de detecção de embarcações possui relevância prática direta em contextos de monitoramento marítimo, segurança costeira e aplicações ambientais. Tradicionalmente, soluções baseadas em arquiteturas profundas, como a família YOLO, alcançam resultados robustos, porém a um custo computacional e energético elevado, o que pode inviabilizar sua aplicação em dispositivos de baixo consumo, como plataformas embarcadas ou sistemas com autonomia limitada. Esse cenário motivou a investigação de alternativas mais leves e acessíveis, capazes de equilibrar desempenho e eficiência.

Para atender a esse objetivo, foi proposto um *pipeline* modular dividido em quatro etapas principais: pré-processamento das imagens, extração de características por meio da MobileNetV3 Small pré-treinada no ImageNet, classificação multiclasse utilizando a rede WiSARD e regressão dos parâmetros das caixas delimitadoras.

Inicialmente, a regressão foi realizada por meio de regressão por vetores de suporte (SVR); posteriormente, o *pipeline* foi estendido para incorporar também abordagens baseadas em k-vizinhos mais próximos (kNN) e uma versão regressiva da WiSARD (ReWiSARD). Essa ampliação permitiu avaliar diferentes compromissos entre custo computacional, consumo energético e desempenho temporal, reforçando o caráter flexível da solução proposta. O modelo YOLOv8 foi adotado como *baseline*, funcionando como referência de comparação devido à sua ampla utilização e desempenho consolidado na literatura.

Durante a implementação, alguns desafios técnicos foram enfrentados. A etapa de pré-processamento mostrou-se a mais custosa em termos de tempo e energia, exigindo operações adicionais como normalização, binarização para adequação à WiSARD e integração das saídas para os modelos regressivos. Além disso, o conjunto de dados, embora anotado em formato compatível com YOLO, precisou ser adaptado para alimentar o *pipeline* proposto, evidenciando a necessidade de compatibilização entre diferentes arquiteturas e representações de dados.

A avaliação experimental foi organizada em quatro dimensões: classificação, regressão, eficiência temporal e eficiência energética. Na tarefa de classificação, a WiSARD apresentou resultados competitivos, com elevada precisão e bom equilíbrio no F1-Score, embora o YOLOv8 tenha atingido valores superiores de revocação. Na etapa de regressão, o SVR demonstrou bom desempenho na estimativa das coordenadas centrais  $(x, y)$ , aproximando-se do modelo profundo de referência, enquanto diferenças mais acentuadas foram observadas nos parâmetros de escala  $(w, h)$ . A inclusão do kNN e da ReWiSARD permitiu ampliar a análise, evidenciando que diferentes métodos regressivos apresentam comportamentos distintos, especialmente no que se refere ao custo de treinamento, tempo de inferência e consumo energético.

Os resultados mais expressivos, entretanto, foram obtidos nas análises de eficiência temporal e energética. O *pipeline* baseado em redes sem pesos apresentou tempos de treinamento e inferência significativamente inferiores aos do YOLOv8, além de consumo energético até duas ordens de grandeza menor. A análise comparativa entre os métodos de regressão demonstrou que a escolha do regressor deve ser orientada pelos requisitos da aplicação: enquanto abordagens como o kNN favorecem cenários com treinamento rápido, a ReWiSARD mostrou-se particularmente atrativa em contextos que priorizam baixo consumo energético durante a inferência contínua em dispositivos embarcados.

Como contribuição científica, este trabalho apresenta o algoritmo denominado **SOD-Edge**, reforçando a aplicabilidade prática das redes neurais sem pesos em tarefas de visão computacional. Os resultados evidenciam que soluções não paramétricas, quando combinadas a técnicas estatísticas leves, podem reduzir significativamente os custos computacionais e energéticos sem comprometer o desempenho

global. Dessa forma, o SOD-Edge abre caminho para o uso mais amplo dessas arquiteturas em cenários de monitoramento embarcado e sistemas de baixa autonomia, contribuindo para a disseminação de abordagens acessíveis, eficientes e sustentáveis na área de visão computacional.

## 6.2 Limitações do Trabalho

Apesar dos resultados promissores, algumas limitações relevantes foram identificadas durante o desenvolvimento e a validação do *pipeline* proposto, as quais devem ser consideradas na interpretação dos resultados e na análise de sua aplicabilidade prática.

A primeira e mais evidente limitação refere-se à detecção de apenas um objeto por imagem. O *pipeline* foi concebido e avaliado em um cenário simplificado, no qual cada imagem contém uma única embarcação. Embora essa configuração tenha permitido validar a proposta de forma controlada e sistemática, ela não reflete plenamente a complexidade de ambientes marítimos reais, nos quais múltiplas embarcações podem coexistir em diferentes distâncias, ângulos e escalas. Essa restrição limita a aplicação direta do modelo em cenários operacionais de monitoramento contínuo.

Outra limitação está associada à regressão dos parâmetros das caixas delimitadoras, em especial aqueles relacionados à escala  $(w, h)$ . Embora o SVR tenha apresentado desempenho satisfatório na estimativa das coordenadas centrais  $(x, y)$ , diferenças mais acentuadas foram observadas nos parâmetros de largura e altura quando comparados ao YOLOv8. A inclusão de métodos alternativos de regressão, como o kNN e a ReWiSARD, evidenciou que não existe uma única abordagem ótima: enquanto o kNN sofre com custos elevados de inferência em função de sua natureza baseada em distância, a ReWiSARD apresenta maior custo de treinamento devido à construção de suas estruturas internas. Esses resultados indicam que abordagens regressivas leves podem enfrentar limitações estruturais na modelagem de variações geométricas complexas, especialmente em cenários com grande diversidade de formatos e proporções entre embarcações.

O conjunto de dados utilizado também constitui uma limitação importante. Composto por menos de mil imagens, sua escala reduzida pode impactar a capacidade de generalização do *pipeline*. Além disso, embora o conjunto apresente alguma variabilidade em termos de ângulos e condições de iluminação, ele não contempla cenários ambientais extremos, como neblina, chuva intensa, reflexos severos da superfície da água ou condições de baixa visibilidade, que são comuns em aplicações reais de monitoramento marítimo. Dessa forma, os resultados obtidos devem ser analisados com cautela, reconhecendo as restrições impostas pela base de dados.

No que se refere à infraestrutura de execução, os experimentos foram conduzidos em dois cenários específicos: um ambiente computacional com GPU (Google Colab Pro) e um dispositivo embarcado de baixo custo (Raspberry Pi 4). Embora representativos, esses ambientes não abrangem toda a diversidade de plataformas embarcadas atualmente disponíveis. Arquiteturas especializadas, como Jetson Nano, Coral TPU ou soluções baseadas em FPGA, não foram consideradas, o que limita a avaliação da portabilidade e do desempenho do *pipeline* em diferentes contextos de hardware.

Por fim, destaca-se a dependência do pré-processamento como um gargalo do sistema. Apesar da eficiência observada nas etapas de classificação e regressão, a extração de características e a normalização dos dados concentraram a maior parcela do custo computacional e energético do pipeline. Essa dependência reduz parcialmente os ganhos obtidos nas etapas subsequentes, sobretudo em dispositivos embarcados com capacidade de processamento limitada.

Essas limitações não invalidam os resultados apresentados, mas delineiam claramente os desafios a serem superados para a aplicação do *pipeline* proposto em cenários reais de monitoramento marítimo. Ao mesmo tempo, elas apontam direções naturais para trabalhos futuros, nos quais extensões para multidetecção, estratégias regressivas mais expressivas e otimizações no pré-processamento podem aproximar ainda mais a abordagem baseada em redes sem pesos da robustez oferecida por arquiteturas profundas.

### 6.3 Trabalhos Futuros

As limitações apontadas nesta dissertação abrem caminhos naturais para trabalhos futuros, que podem contribuir tanto para o amadurecimento da estratégia híbrida de detecção proposta quanto para o avanço da área de visão computacional embarcada como um todo.

O primeiro e mais relevante desdobramento é a implementação de uma solução de multidetecção. A adaptação do *pipeline* para lidar com múltiplos objetos por imagem exige mudanças estruturais, tanto na etapa de classificação quanto na de regressão. Do ponto de vista da WiSARD, seria necessário empregar estratégias de segmentação ou varredura por janelas que permitam classificar diversas regiões de uma mesma imagem. No que se refere à regressão, embora diferentes abordagens já tenham sido investigadas neste trabalho — incluindo SVR, kNN e ReWiSARD —, a extensão para múltiplas instâncias regressivas, associadas a diferentes regiões da imagem, permanece como um desafio em aberto. Essa evolução é fundamental para aproximar a proposta das condições reais de monitoramento marítimo, nas quais múltiplas embarcações podem coexistir em um mesmo quadro.

Um segundo desdobramento natural consiste na expansão da abordagem para dados em vídeo. O processamento de sequências temporais possibilita novas funcionalidades, como o rastreamento contínuo de embarcações, a estimação de trajetórias e a identificação de padrões anômalos de movimento. Para tanto, seria necessário integrar algoritmos de *tracking*, como SORT ou DeepSORT, ao *pipeline* existente, preservando a leveza da proposta e evitando o aumento excessivo do custo computacional e energético. Tal avanço ampliaria o escopo de aplicação do modelo para cenários de vigilância e monitoramento em tempo real.

Outra linha de continuidade refere-se à otimização da etapa de pré-processamento, identificada ao longo dos experimentos como o principal gargalo do sistema. Estratégias como a redução da dimensionalidade das *features*, o uso de quantização mais agressiva ou a substituição da MobileNetV3 por extratores ainda mais compactos — como variantes *lite* ou arquiteturas especificamente projetadas para dispositivos móveis — podem reduzir significativamente o custo temporal e energético dessa etapa, potencializando ainda mais os ganhos do *pipeline* proposto.

No que diz respeito aos ambientes de execução, torna-se relevante a avaliação do *pipeline* em outras plataformas embarcadas. Testes em dispositivos como Jetson Nano, Coral TPU ou soluções baseadas em FPGA permitiriam uma análise mais ampla da portabilidade, do desempenho e do consumo energético do modelo em diferentes arquiteturas de hardware. Essa investigação contribuiria para verificar se os ganhos observados no Raspberry Pi 4 se mantêm em plataformas amplamente utilizadas em aplicações de visão computacional na borda.

Por fim, destaca-se a importância de explorar conjuntos de dados mais amplos e diversificados. A inclusão de bases com maior número de imagens e que contemplem condições ambientais adversas, como neblina, chuva intensa, reflexos severos da superfície da água e variações significativas de iluminação, proporcionaria uma avaliação mais robusta do *pipeline*. Essa ampliação permitiria investigar com maior profundidade a capacidade de generalização do modelo e identificar limitações em cenários mais próximos da realidade operacional.

Essas direções de pesquisa não apenas endereçam as limitações identificadas ao longo do trabalho, mas também ampliam o escopo da proposta, consolidando o *pipeline* baseado em redes neurais sem pesos e regressão leve como uma solução promissora para aplicações práticas de monitoramento marítimo em dispositivos de baixo consumo.

## 6.4 Considerações Finais

O desenvolvimento deste trabalho permitiu demonstrar que soluções leves e baseadas em aprendizado não paramétrico podem constituir alternativas viáveis a arquitetu-

ras profundas em tarefas de visão computacional embarcada. A proposta do *pipeline* baseado em redes neurais sem pesos e regressão leve demonstrou que é possível alcançar resultados competitivos em classificação e regressão de embarcações, ao mesmo tempo em que se obtêm ganhos expressivos em eficiência temporal e energética, fatores críticos em contextos de recursos limitados.

Ao longo da dissertação, buscou-se evidenciar que a adoção de modelos tradicionais, muitas vezes negligenciados em função da popularidade de arquiteturas profundas, ainda pode oferecer contribuições relevantes quando aliada a estratégias modernas de extração de características e codificação de dados. A combinação entre a MobileNetV3 Small, a rede WiSARD e diferentes abordagens regressivas — incluindo SVR, kNN e ReWiSARD — resultou em uma solução híbrida, de baixo custo computacional, capaz de competir com um modelo de referência consolidado na literatura, como o YOLOv8.

Do ponto de vista experimental, os resultados mostraram que, enquanto o YOLOv8 se destacou na revocação e na regressão de escala, o *pipeline* proposto obteve maior precisão na etapa de classificação e desempenho competitivo na estimativa da localização central das embarcações. A avaliação comparativa entre os métodos regressivos evidenciou ainda que não existe uma única solução ótima, mas sim diferentes compromissos entre custo de treinamento, tempo de inferência e consumo energético. Entretanto, os maiores diferenciais da proposta estiveram nas métricas de eficiência: treinamento quase instantâneo para alguns modelos, inferência em tempo compatível com aplicações em tempo real e consumo energético até duas ordens de grandeza inferior ao *baseline*. Esses achados reforçam a adequação do modelo para cenários embarcados e aplicações em campo, como monitoramento marítimo com drones ou dispositivos de baixa autonomia energética.

Do ponto de vista científico, este trabalho contribui ao ampliar a discussão sobre o papel de redes neurais sem pesos em aplicações práticas de visão computacional. Ao reposicionar a WiSARD, tradicionalmente explorada em contextos educacionais ou experimentais, em um *pipeline* voltado à detecção embarcada, demonstrou-se que a simplicidade estrutural pode se traduzir em eficiência e viabilidade prática. Além disso, a integração com diferentes estratégias de regressão mostrou-se fundamental para complementar a etapa de classificação com capacidades de predição contínua, fortalecendo a proposta como uma solução flexível e adaptável a diferentes requisitos de aplicação.

Por fim, este trabalho reforça a ideia de que a evolução da visão computacional não depende exclusivamente do aumento da complexidade das arquiteturas profundas. Em muitos cenários, sobretudo aqueles em que os recursos de *hardware* e energia são limitados, a adoção de soluções simples, bem estruturadas e energeticamente eficientes pode representar um caminho mais sustentável e acessível. Assim,

os resultados aqui obtidos contribuem não apenas para a literatura de detecção de embarcações, mas também para o avanço da pesquisa em modelos alternativos, abrindo espaço para aplicações práticas em contextos até então restritos pela elevada demanda computacional.

# Referências Bibliográficas

- SANTIAGO, L., VERONA, L., RANGEL, F., et al. “Weightless neural networks as memory segmented bloom filters”, *Neurocomputing*, v. 416, pp. 292–304, 2020.
- ROSENBAUM, L., DÖRR, A., BAUER, M., et al. “Inferring multi-target QSAR models with taxonomy-based multi-task learning”, *Journal of cheminformatics*, v. 5, pp. 33, 07 2013. doi: 10.1186/1758-2946-5-33.
- ZHAO, T., WANG, Y., XU, Q., et al. “Ship detection with deep learning in optical remote-sensing images: A survey of challenges and advances”, *Remote Sensing*, v. 16, n. 7, pp. 1145, 2024. doi: 10.3390/rs16071145. Disponível em: <<https://www.mdpi.com/2072-4292/16/7/1145>>.
- MOOSBAUER, M., ZAMBANINI, S., KAMPEL, M. “A Benchmark for Deep Learning Based Object Detection in Maritime Environments”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 0–0, 2019. Disponível em: <[https://openaccess.thecvf.com/content\\_CVPRW\\_2019/html/PBVS/Moosbauer\\_A\\_Benchmark\\_for\\_Deep\\_Learning\\_Based\\_Object\\_Detection\\_in\\_Maritime\\_CVPRW\\_2019\\_paper.html](https://openaccess.thecvf.com/content_CVPRW_2019/html/PBVS/Moosbauer_A_Benchmark_for_Deep_Learning_Based_Object_Detection_in_Maritime_CVPRW_2019_paper.html)>.
- ZHU, B., WANG, Y., LI, Q. “Ship Detection Using Edge-Based Segmentation and Histogram of Oriented Gradient with Ship Size Ratio”, *Proceedings of the International Conference on Image Processing*, 2016.
- ALVEAR-SANDOVAL, R., DEL CAMPO, D., IZQUIERDO, E. “Ship detection from coastal surveillance videos via an ensemble Canny-Gaussian-morphology framework”, *Journal of Navigation*, v. 70, n. 5, pp. 1041–1060, 2017.
- DALAL, N., TRIGGS, B. “Histograms of oriented gradients for human detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 886–893, 2005.

- LOWE, D. G. “Distinctive image features from scale-invariant keypoints”, *International Journal of Computer Vision*, v. 60, n. 2, pp. 91–110, 2004.
- HU, W., TAN, T., WANG, L., et al. “A survey on visual surveillance of object motion and behaviors”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, v. 34, n. 3, pp. 334–352, 2004.
- WANG, L., ZHAO, H., ZHANG, Y. “A review of intelligent ship marine object detection based on RGB camera”, *IET Image Processing*, v. 17, n. 7, pp. 1687–1702, 2023.
- LECUN, Y., BENGIO, Y., HINTON, G. “Deep learning”, *Nature*, v. 521, n. 7553, pp. 436–444, 2015. doi: 10.1038/nature14539.
- KRIZHEVSKY, A., SUTSKEVER, I., HINTON, G. E. “ImageNet classification with deep convolutional neural networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*, v. 25, 2012. Disponível em: <<https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>>.
- REN, S., HE, K., GIRSHICK, R., et al. “Faster R-CNN: Towards real-time object detection with region proposal networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 91–99, 2015.
- LIU, W., ANGUELOV, D., ERHAN, D., et al. “Ssd: Single shot multibox detector”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21–37. Springer, 2016.
- REDMON, J., DIVVALA, S., GIRSHICK, R., et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.
- MURSHED, M. G. S., MURPHY, C., HOU, D., et al. “Machine Learning at the Network Edge: A Survey”, *arXiv preprint arXiv:1908.00080*, 2019.
- CHEN, Q., ZHU, H., YANG, L., et al. “Edge Computing Assisted Autonomous Flight for UAV: Synergies between Vision and Communications”, *arXiv preprint arXiv:2012.05517*, 2020.

- SORET, B., NGUYEN, L. D., SEEGER, J., et al. “Learning, Computing, and Trustworthiness in Intelligent IoT Environments: Performance–Energy Tradeoffs”, *arXiv preprint arXiv:2110.01686*, 2021.
- SINGH, R., OTHERS. “Edge AI: A survey”, *Journal of Internet of Things / Edge Computing (ou equivalente)*, 2023. survey sobre conceitos de Edge AI.
- SURIANARAYANAN, C., OTHERS. “A Survey on Optimization Techniques for Edge Artificial Intelligence”, *PMC (ou nome do periódico, se aplicável)*, 2023. técnicas para otimização de modelos em edge.
- WANG, X., TANG, Z., GUO, J., et al. “Empowering Edge Intelligence: A Comprehensive Survey on On-Device AI Models”, *arXiv preprint arXiv:2503.06027*, 2025a. survey recente sobre modelos on-device.
- LIU, W., ZHANG, H., CHEN, F. “SDNet: a lightweight ship detection network in remote sensing images by super-resolution enhancement and detail completion”, *European Journal of Remote Sensing*, v. 58, n. 1, pp. 1–20, 2025. doi: 10.1080/22797254.2025.2469863. Disponível em: <<https://www.tandfonline.com/doi/full/10.1080/22797254.2025.2469863>>.
- HOWARD, A. G., ZHU, M., CHEN, B., et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”, *arXiv preprint arXiv:1704.04861*, 2017.
- IANDOLA, F. N., HAN, S., MOSKEWICZ, M. W., et al. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size”, *arXiv preprint arXiv:1602.07360*, 2016.
- ABADADE, Y., TEMOUDEN, A., BAMOUMEN, H., et al. “A comprehensive survey on tinyml”, *IEEE Access*, v. 11, pp. 96892–96922, 2023.
- ALEKSANDER, I., THOMAS, W., BOWDEN, P. “WISARD: a radical step forward in image recognition”, *Sensor review*, v. 4, n. 3, pp. 120–124, 1984.
- PEREIRA, A. G., DE FARIAS, C. M., DE ARAUJO, L. S. “A Ship Detection Technique Using Weightless Neural Networks”. In: *2025 28th International Conference on Information Fusion (FUSION)*, pp. 1–8. IEEE, 2025.
- SINGH, R., GILL, S. S. “Edge AI: a survey”, *Internet of Things and Cyber-Physical Systems*, v. 3, pp. 71–92, 2023.

- ZHANG, R., DONG, M., LIU, Y., et al. “Survey on deep learning-based marine object detection”, *Computational Intelligence and Neuroscience*, v. 2021, pp. 1–16, 2021. doi: 10.1155/2021/5808206. Disponível em: <<https://doi.org/10.1155/2021/5808206>>.
- ZHAO, C., LIU, Y., SUN, X., et al. “Deep Learning-Based Object Detection in Maritime UAV Imagery: Review and Experimental Comparisons”, *arXiv preprint arXiv:2311.07955*, 2023. Disponível em: <<https://arxiv.org/abs/2311.07955>>.
- LECUN, Y., BOTTOU, L., BENGIO, Y., et al. “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, v. 86, n. 11, pp. 2278–2324, 1998. doi: 10.1109/5.726791.
- HE, K., ZHANG, X., REN, S., et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.
- SIMONYAN, K., ZISSERMAN, A. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556*, 2014. Disponível em: <<https://arxiv.org/abs/1409.1556>>.
- SANDLER, M., HOWARD, A., ZHU, M., et al. “MobileNetV2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 4510–4520, 2018. doi: 10.1109/CVPR.2018.00474.
- HOWARD, A., SANDLER, M., CHU, G., et al. “Searching for MobileNetV3”. In: *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pp. 1314–1324, 2019. doi: 10.1109/ICCV.2019.00140.
- PAN, S. J., YANG, Q. “A survey on transfer learning”, *IEEE Transactions on knowledge and data engineering*, v. 22, n. 10, pp. 1345–1359, 2009. doi: 10.1109/TKDE.2009.191.
- TAN, M., LE, Q. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.
- DENG, J., DONG, W., SOCHER, R., et al. “ImageNet: A large-scale hierarchical image database”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

- GRIECO, B. P. A., LIMA, P. M. V., DE GREGORIO, M., et al. “Producing pattern examples from “mental” images”, *Neurocomputing*, v. 73, n. 7–9, pp. 1057–1064, 2010. doi: 10.1016/j.neucom.2009.11.015.
- CARVALHO, D. S., CARNEIRO, H. C. C., FRANÇA, F. M. G., et al. “Bleaching: Agile Overtraining Avoidance in the WiSARD Weightless Neural Classifier”. In: *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, 2013.
- LUSQUINO FILHO, L. A., OLIVEIRA, L. F., LIMA FILHO, A., et al. “Extending the weightless WiSARD classifier for regression”, *Neurocomputing*, v. 416, pp. 280–291, 2020.
- VAPNIK, V. *The Nature of Statistical Learning Theory*. Springer, 1995. doi: 10.1007/978-1-4757-2440-0.
- DRUCKER, H., BURGESS, C. J., KAUFMAN, L., et al. “Support Vector Regression Machines”. In: *Advances in Neural Information Processing Systems (NeurIPS)*, v. 9, pp. 155–161, 1997. Disponível em: <<https://proceedings.neurips.cc/paper/1996/hash/d38901788c533e8286cb6400b40b386d-Abstract.html>>.
- SMOLA, A. J., SCHÖLKOPF, B. “A tutorial on support vector regression”, *Statistics and Computing*, v. 14, n. 3, pp. 199–222, 2004. doi: 10.1023/B:STCO.0000035301.49549.88.
- DE GREGORIO, M., GIORDANO, M. “The WiSARD Classifier”. In: *Proceedings of ESANN*, 2016. Disponível em: <<https://www.esann.org/sites/default/files/proceedings/legacy/es2016-63.pdf>>.
- MIRANDA, I. D., ARORA, A., SUSSKIND, Z., et al. “LogicWiSARD: Memoryless Synthesis of Weightless Neural Networks”. In: *2022 IEEE 33rd International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pp. 19–26, 2022. doi: 10.1109/ASAP54787.2022.00014.
- SUSSKIND, Z., ARORA, A., MIRANDA, I. D., et al. “Weightless neural networks for efficient edge inference”. In: *Proceedings of the International Conference on Parallel Architectures and Compilation Techniques*, pp. 279–290, 2022.
- LIMA FILHO, A. S., GUARISA, G. P., LUSQUINO FILHO, L. A., et al. “wisardpkg—A library for WiSARD-based models”, *arXiv e-prints*, pp. arXiv–2005, 2020.

- MILHOMEM, S., ALMEIDA, T. D. S., GRAMACHO, W., et al. “Weightless Neural Network with Transfer Learning to Detect Distress in Asphalt”, *ArXiv preprint arXiv:1901.03660*, 2019. Disponível em: <<https://arxiv.org/abs/1901.03660>>.
- AGARWAL, A., TRIGGS, B. “Recovering human body configuration: Combining segmentation and recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, v. 2, pp. 44–51, 2006. doi: 10.1109/CVPR.2006.252.
- GUO, G., FU, Y. “Human age estimation using bio-inspired features”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008. doi: 10.1109/CVPR.2008.4587583.
- TUIA, D., CAMPS-VALLS, G., MATASCI, G., et al. “Multioutput support vector regression for remote sensing biophysical parameter estimation”, *IEEE Transactions on Geoscience and Remote Sensing*, v. 49, n. 8, pp. 3050–3062, 2011. doi: 10.1109/TGRS.2011.2120615.
- WANG, Y., SUN, C. “Reconstruction of Freeform Surface by Support Vector Regression”. In: *International Conference on Computational Science and Its Applications (ICCSA)*, pp. 305–314. Springer, 2005. doi: 10.1007/11424925\_32.
- ZHANG, S., LI, J., XU, C. “R-CNN-Based Ship Detection from High Resolution Remote Sensing Imagery”. In: *Proceedings of the International Conference on Image Processing (ICIP)*, 2019.
- LI, B., SUN, H., TANG, H. “Ship detection and classification from optical remote sensing images”, *Journal of Systems Engineering and Electronics*, v. 32, n. 1, pp. 202–214, 2021. doi: 10.23919/JSEE.2021.000018. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1000936120304544>>.
- WANG, L., LIU, Z., CHEN, X. “YOLO-ssboat: An Improved YOLO for Small Ship Detection in Complex Maritime Environments”, *Remote Sensing*, v. 17, n. 11, pp. 1948, 2025b. doi: 10.3390/rs17111948. Disponível em: <<https://www.mdpi.com/2072-4292/17/11/1948>>.
- XIAO, K., WANG, L., CHEN, M. “SAR Ship Detection based on Swin Transformer and Feature Enhancement Feature Pyramid Network”, *Remote Sensing*, v. 14, n. 15, pp. 3621, 2022.

- BETTI, A., PIERI, G., CORSINI, G. “Real-time target detection in maritime scenarios based on YOLOv3 model”. In: *OCEANS 2020: IEEE Conference*, pp. 1–7, 2020.
- WONG, A., FAMUORI, M., SHAFIEE, M. J., et al. “YOLO nano: A highly compact you only look once convolutional neural network for object detection”. In: *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*, pp. 22–25. IEEE, 2019.
- KUZMIC, J., RUDOLPH, G. “Object Detection with TensorFlow on Hardware with Limited Resources for Low-power IoT Devices.” In: *IJCCI*, pp. 302–309, 2021.