UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

MATEUS SCHEIN CAVALHEIRO CORRÊA -
00311352

DESENVOLVIMENTO DE SISTEMA PARA MONITORAMENTO E ADESTRAMENTO DE ANIMAIS DOMÉSTICOS BASEADO EM VISÃO COMPUTACIONAL

Porto Alegre
2026

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

MATEUS SCHEIN CAVALHEIRO CORRÊA -
00311352

DESENVOLVIMENTO DE SISTEMA PARA MONITORAMENTO E ADESTRAMENTO DE ANIMAIS DOMÉSTICOS BASEADO EM VISÃO COMPUTACIONAL

Trabalho de Conclusão de Curso submetido à COMGRAD/CCA da UFRGS como parte dos requisitos para a obtenção do título de *Bacharel em Engenharia de Controle e Automação.*

Orientadores:
Prof. Dr. Heraldo José de Amorim

Eng. Jonnas Espindula

Porto Alegre
2026

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

MATEUS SCHEIN CAVALHEIRO CORRÊA -
00311352

DESENVOLVIMENTO DE SISTEMA PARA MONITORAMENTO E ADESTRAMENTO DE ANIMAIS DOMÉSTICOS BASEADO EM VISÃO COMPUTACIONAL

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção dos créditos da Atividade de *Trabalho de Conclusão de Curso CCA - II* e aprovado em sua forma final pelos Orientadores e Banca Examinadora abaixo.

Orientadores:
Prof. Dr. Heraldo José de Amorim, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Eng. Jonnas Espindula, UFRGS
Bacharel pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Banca Examinadora:

Prof. Dr. Heraldo José de Amorim, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Marcelo Götz, UFRGS
Doutor pela Universität Paderborn – Paderborn, Alemanha

Prof. Dr. Edson Cordeiro do Valle, UFRGS
Doutor pela Universidade Federal da Bahia – Salvador, Brasil

Fabiano Disconzi Wildner
Coordenador de Curso
Eng. de Controle e Automação

Porto Alegre
Julho - 2026

Agradeço à minha família, que me apoiou durante a graduação.

Ao meu grande amor, pela paciência e compreensão durante o período de escrita deste trabalho.

Ao meu gato, Frajola, que me auxiliou durante o trabalho, tanto na metodologia, como modelo, quanto na escrita, me fazendo companhia por horas a fio.

Ao Grupo de Apoio. Acho difícil descrevê-lo, mas palavras como união, persistência e companheirismo parecem se aconchegar no que seria sua forma. Queridos amigos, sempre fico muito feliz em olhar pra trás e ver nossa jornada juntos. Jamais teria conseguido sem vocês.

Agradeço a todos que fizeram parte da minha jornada na graduação, dentro e fora do campus. Criei laços em diversos lugares desse imenso universo que chamamos de faculdade, que de maneira direta e indireta me ajudaram a perseverar nos momentos mais difíceis. Nem só de estudo se faz um curso. O convívio com pessoas tão especiais despertou em mim hobbies e atividades que foram cruciais para me manter inteiro, como corrida, cerâmica e pedal. Vocês sabem quem são. Muito obrigado.

Projeto Mosquitoramento, Equipe de Robótica Startche, LAMECC e USRL, muito obrigado por me proporcionarem tantas oportunidades, graças a vocês pude ir a lugares que jamais imaginei que iria e estudar temas que se tornaram minha paixão.

Bateria Minotrago, sou muito grato pelas noites de festa que tive a honra de tocar com vocês. Espero continuar ouvindo seu coração bater sempre que passar pela Redenção aos fins de semana.

Aos meus orientadores, Heraldo e Jonnas, que não mediram esforços durante a jornada do desenvolvimento desse projeto.

À Universidade Federal do Rio Grande do Sul, UFRGS, pela oportunidade de realização de estudos.

Apressa-te a viver bem e pensa que cada dia é, por si só, uma vida.

Sêneca

RESUMO

Soluções existentes de monitoramento e adestramento de animais domésticos atuam predominantemente de forma passiva, com pouca incorporação de detecção contextual e resposta autônoma ao comportamento do animal. Este trabalho propõe o desenvolvimento de um sistema automatizado de adestramento para animais domésticos baseado em visão computacional e descreve seu desenvolvimento. O objetivo é identificar, através de detecção de imagens, a presença do animal em regiões do ambiente definidas como restritas e emitir um estímulo sonoro de forma autônoma, previsível e sem causar qualquer dano físico, buscando demonstrar a viabilidade técnica do sistema embarcado desenvolvido para auxiliar na modificação de comportamentos indesejados. A definição das regiões de interesse (Regions of Interest, ROIs) é realizada pelo usuário por meio de uma interface web hospedada no próprio dispositivo, permitindo configurar zonas de monitoramento diretamente sobre a imagem da câmera. O sistema integra algoritmos de detecção de objetos, verificação geométrica da presença do animal em cada ROI e controle de um emissor sonoro com parâmetros de frequência e duração ajustáveis conforme a eficácia observada. A revisão bibliográfica abrange conceitos de comportamento animal, visão computacional, detecção espacial e controle auditivo. Os resultados obtidos demonstram operação autônoma em tempo de execução a 3,22 FPS com latência total de 310 ms, recall de $90,2\%$ no cenário binário unificado e taxa residual de falsos positivos inferior a 3% sobre o fundo estático, validando a viabilidade técnica da solução embarcada proposta.

Palavras-chave: Automação Residencial, Comportamento Animal, Sistemas Embarcados, Visão Computacional

ABSTRACT

Existing solutions for monitoring and training domestic animals act predominantly in a passive manner, with little incorporation of contextual detection and autonomous response to the animal's behavior. This work proposes an automated training system for domestic animals based on computer vision and describes its development process. The goal is to detect, through a camera, the presence of the animal in user-defined restricted areas and automatically emit a sound stimulus that is predictable, harmless, and non-aversive, aiming to demonstrate the technical feasibility of an embedded system that combines visual perception and adaptive auditory response to assist in modifying undesired behaviors. The regions of interest (ROIs) are configured via a web interface hosted on the device itself, allowing the user to define monitoring zones directly on the camera image. The system integrates object detection algorithms, geometric verification of the animal's presence within each ROI, and control of a sound emitter with frequency and duration parameters adjustable according to behavioral feedback. The theoretical review covers animal behavior, computer vision, spatial detection, and sound control. The results demonstrate autonomous execution time operation at 3.22 FPS with a total end-to-end latency of 310 ms, 90.2% recall in the unified binary scenario, and a residual false positive rate of approximately 3% on static background, validating the technical feasibility of the proposed embedded solution.

Palavras-chave: Animal Behavior, Computer Vision, Embedded Systems, Residential Automation

LISTA DE ILUSTRAÇÕES

1 Algoritmo descritivo do sistema. 19
2 Proposta do diagrama de blocos da topologia distribuída do sistema: fluxo de dados bidirecional entre o nó coletor, o nó processador e a interface web. 26
3 Exemplo de amostra sintética gerada pelo *pipeline* de colagem contextual através de remoção de fundo e sobreposição, integrando o objeto de interesse ao room-dataset.Fonte: Autor 29
4 Diagrama de blocos da topologia distribuída do sistema: fluxo de dados bidirecional entre o ESP32, a Raspberry Pi 4 e a interface web. 31
5 Interface gráfica do usuário evidenciando o *banner* de alerta, a demarcação da Região de Interesse (ROI), a *bounding box* sobre o animal detectado e o painel de controle lateral. Fonte: Autor 36
6 Diagrama de conexões elétricas e interfaceamento lógico do subsistema de atuação acústica. 37
7 Curvas de convergência do treinamento: (A) evolução das perdas e (B) métricas de validação ao longo das épocas. Fonte: Autor 40
8 Matriz de confusão normalizada para o cenário binário unificado. Fonte: Autor 41
9 Matriz de confusão normalizada para o cenário multiclasse (cão, gato e background). Fonte: Autor 43
10 Amostras experimentais do *pipeline* de inferência do modelo YOLOv8n. Fonte: Elaborado pelo Autor sobre frames extraídos de The Pet Collective (2025) . 44
11 Interface de monitoramento evidenciando a robustez da sinalização de invasão perimetral frente a variações probabilísticas de classe. Fonte: Autor 46
12 Circuito eletrônico de potência e integração dos atuadores no ESP32. Fonte: Autor 47
13 Curva de operação do sistema: Posição do servo motor versus Frequência do sinal de áudio. Fonte: Autor 48

LISTA DE TABELAS

1 Distribuição quantitativa e proporcional das amostras do dataset final. 28
2 Análise comparativa de alternativas para o nó periférico de aquisição. . 32
3 Análise comparativa de alternativas para o nó central de inferência. . . . 32
4 Métricas de desempenho consolidadas para a estrutura binária unificada. 41
5 Métricas de desempenho estatístico discriminadas por classes individuais. 42
6 Decomposição final dos tempos de latência e taxa de quadros obtidos nos ensaios de bancada. 45

LISTA DE SÍMBOLOS

$\cap$	Interseção entre conjuntos
$\cup$	União entre conjuntos
IoU	Interseção sobre União, $\text{IoU}(A, B) = \dfrac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$
$(x_i,\, y_i)$	Coordenadas do i-ésimo vértice de um polígono no sistema de referência da imagem
$M(x)$	Máscara binária de segmentação do animal (1 no objeto, 0 no fundo)
$G(x)$	Imagem sintética final gerada pelo pipeline de sobreposição contextual
$I_{\text{animal}}(x)$	Imagem de origem contendo o animal segmentado
$I_{\text{cenario}}(x)$	Imagem de cenário residencial do room-dataset
lr_0	Taxa de aprendizado inicial do otimizador
mAP_{50}	Precisão média no limiar IoU = 0,50
mAP_{50-95}	Precisão média nos limiares IoU de 0,50 a 0,95
F1	F1-score, média harmônica entre precisão e revocação
t_0	Instante inicial de acionamento do atuador acústico
$\mathcal{O}(1)$	Complexidade computacional constante
$\mathcal{O}(N \cdot M)$	Complexidade computacional proporcional ao produto do número de vértices dos polígonos
$\mathcal{O}(N)$	Complexidade computacional linear no número de vértices
μ**s**	Microssegundo (10^{-6} s)
ms	Milissegundo (10^{-3} s)
m/s	Metro por segundo (unidade de velocidade)
m	Metro (unidade de comprimento)

W	Watt (unidade de potência)
$^\circ$	Grau (unidade de ângulo)
V	Volt (unidade de tensão elétrica)

LISTA DE ACRÔNIMOS

ABNT	Associação Brasileira de Normas Técnicas
API	Application Programming Interface (Interface de Programação de Aplicações)
BLE	Bluetooth Low Energy (Bluetooth de Baixo Consumo)
CNN	Convolutional Neural Network (Rede Neural Convolucional)
COCO	Common Objects in Context (Objetos comuns no contexto)
COM	Component Object Model (Modelo de Objeto de Componente)
CPU	Central Processing Unit (Unidade Central de Processamento)
CSS	Cascading Style Sheets (Folhas de Estilo em Cascata)
dB	Decibel
DOM	Document Object Model (Modelo de Objeto de Documento)
EKF	Extended Kalman Filter (Filtro de Kalman Estendido)
ESP32	Microcontrolador Wi-Fi/BLE da Espressif
F1	F1-score, média harmônica entre precisão e revocação
FIFO	First In, First Out (Primeiro a Entrar, Primeiro a Sair)
FN	Falso Negativo (instância real não detectada pelo modelo)
FP	Falso Positivo (detecção incorretamente classificada)
FPS	Frames Per Second (Quadros por Segundo)
FPPI	False Positives Per Image (Falsos Positivos por Imagem)
GPIO	General Purpose Input/Output (Entrada/Saída de Propósito Geral)
GPU	Graphics Processing Unit (Unidade de Processamento Gráfico)
HOG	Histogram of Oriented Gradients (Histograma de Gradientes Orientados)
HSV	Hue, Saturation, Value (Matiz, Saturação, Valor)
HTTP	HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto)
IEEE	Institute of Electrical and Electronics Engineers
Hz	Hertz
I2C	Inter-Integrated Circuit (Barramento serial I2C)

IoT	Internet of Things (Internet das Coisas)
IoU	Intersection over Union (Interseção sobre União)
JPEG	Joint Photographic Experts Group (formato de compressão de imagem)
JSON	JavaScript Object Notation (Notação de Objeto JavaScript)
LED	Light-Emitting Diode (Diodo Emissor de Luz)
LoRa	Long Range (modulação/PHY de longo alcance usada em LPWAN)
mAP	mean Average Precision (Precisão Média)
mAP50	Precisão média no limiar IoU = 0,50
mAP50-95	Precisão média nos limiares IoU de 0,50 a 0,95
M-JPEG	Motion JPEG (formato de vídeo por sequência de quadros JPEG)
MQTT	Message Queuing Telemetry Transport (Protocolo de Telemetria por Fila de Mensagens)
NO	Normally Open (Normalmente Aberto)
NPU	Neural Processing Unit (Unidade de Processamento Neural)
ONNX	Open Neural Network Exchange (formato aberto de intercâmbio de redes neurais)
OpenCV	Open Source Computer Vision Library (Biblioteca de Visão Computacional de Código Aberto)
PWM	Pulse Width Modulation (Modulação por Largura de Pulso)
QVGA	Quarter Video Graphics Array (resolução 320×240 pixels)
R-CNN	Region-based Convolutional Neural Network (Rede Neural Convolucional Baseada em Regiões)
RAM	Random Access Memory (Memória de Acesso Aleatório)
RGB	Red, Green, Blue (Vermelho, Verde, Azul — modelo de cor aditivo)
ROI	Region of Interest (Região de Interesse)
RTC	Real-Time Clock (Relógio de Tempo Real)
SIFT	Scale-Invariant Feature Transform (Transformada de Características Invariante à Escala)
SPI	Serial Peripheral Interface (Interface Serial Periférica)
SPL	Sound Pressure Level (Nível de Pressão Sonora)

TinyML	Tiny Machine Learning (Aprendizado de Máquina Embarcado em Dispositivos de Baixo Recurso)
TCP/IP	Transmission Control Protocol/Internet Protocol (Protocolo de Controle de Transmissão/Protocolo de Internet)
ToF	Time of Flight (Tempo de Voo)
TPU	Tensor Processing Unit (Unidade de Processamento Tensorial)
UART	Universal Asynchronous Receiver/Transmitter (Receptor/Transmissor Assíncrono Universal)
UKF	Unscented Kalman Filter (Filtro de Kalman Não Linear)
URL	Uniform Resource Locator (Localizador Uniforme de Recursos)
VGG	Visual Geometry Group (arquitetura de CNN desenvolvida pelo grupo VGG de Oxford)
VP	Verdadeiro Positivo (detecção corretamente classificada)
Wi-Fi	Wireless Fidelity (Rede sem fio — IEEE 802.11)
XCLK	External Clock (Sinal de Relógio Externo para módulos de câmera)
XML	Extensible Markup Language (Linguagem de Marcação Extensível)
YOLO	You Only Look Once (algoritmo de detecção de objetos em tempo real)
YCbCr	Luminância e Crominância (modelo de cor para compressão de vídeo/imagem)
YUV	Luminância e Crominância UV (modelo de cor analógico/digital)

SUMÁRIO

1	Introdução	17
2	Revisão bibliográfica e Estado da Arte	18
2.1	Comportamento Animal	18
2.2	Visão Computacional	19
2.3	Detecção e Definição de Regiões de Interesse (ROI)	21
2.4	Fontes Sonoras e Estímulos Condicionados	22
2.5	Sistemas de Monitoramento Automatizado	23
2.6	Processamento em tempo de execução em sistemas de visão embarcados	24
3	Desenvolvimento do Projeto	26
3.1	Desenvolvimento da Rede Neural de Detecção	26
3.1.1	Seleção da Arquitetura do Modelo	26
3.1.2	Composição e Estruturação do Dataset	27
3.1.3	Técnicas de Aumento de Dados (Data Augmentation) e Sobreposição Contextual	28
3.1.4	Estratégia de Treinamento e Validação Prática	30
3.1.5	Topologia Distribuída e Análise Comparativa de Hardware	31
3.1.5.1	Análise do Nó Periférico de Aquisição: ESP32-CAM e Alternativas Microcontroladas	31
3.1.5.2	Análise do Nó Processador Central: Raspberry Pi 4 e Alternativas Single-Board	32
3.1.6	Protocolo de Comunicação e Arquitetura de Rede	33
3.1.7	Desenvolvimento da Interface Web e Pipeline de Sinalização	35
3.2	Mecanismo de Atuação Acústica e Malha de Controle Realimentada	37
3.2.1	Circuito de Chaveamento por Relé e Isolamento de Potência	37
3.2.2	Mecanismo Eletromecânico de Ajuste de Frequência e Controle Adaptativo	38
4	RESULTADOS E DISCUSSÃO	39
4.1	Resultados e Validação do Subsistema de Visão Computacional	39
4.1.1	Otimização Arquitetural: Cenário Binário (Animais Domésticos)	41
4.1.2	Análise Estatística por Classes Individuais	42
4.1.3	Análise Dinâmica e Casos Críticos de Inferência	42

4.2 Desempenho de Rede, Latência e Taxa de Quadros (FPS) 45
4.3 Integração dos Atuadores e Lógica de Acionamento. 47
4.4 Avaliação Comportamental do Subsistema Acústico 48
4.4.1 Funcionamento Observado da Malha de Controle 49
4.4.2 Limitações Identificadas e Análise das Causas 49
5 CONCLUSÃO 51
Referências 53

1 INTRODUÇÃO

O convívio entre seres humanos e animais domésticos tornou-se parte integrante da rotina de grande parcela da população, especialmente em ambientes urbanos. Essa tendência é reflexo do aumento da população de pets (ABINPET, 2022) e da humanização das relações com esses animais (WORLD ANIMAL PROTECTION, 2022). Com isso, surgem também desafios relacionados ao comportamento desses animais dentro das residências, como o acesso a locais inadequados, lixeiras, bancadas, mesas e jardins, que podem causar acidentes, contaminações ou danos a objetos, o que frequentemente envolve a necessidade de adestramento. Métodos convencionais de treinamento baseiam-se em reforço positivo e negativo, aplicados pelo tutor de forma manual e presencial (LINDSAY, 2000). Essa dependência constante da intervenção humana, entretanto, torna o processo inconsistente e de eficácia variável, especialmente quando o tutor não está presente.

O avanço da eletrônica embarcada e da visão computacional impulsionou o desenvolvimento de sistemas capazes de reconhecer e reagir a eventos do ambiente em tempo real. Aplicações como alimentadores automáticos (PETSAFE, 2025), sistemas de vigilância doméstica (ARLO, 2025) e colares eletrônicos de adestramento (GARMIN, 2025) ilustram esse progresso. Contudo, observa-se que a maioria dessas soluções atua de maneira passiva, apenas registrando eventos ou executando tarefas agendadas, e raramente incorpora mecanismos de detecção contextual, resposta autônoma e adaptação ao comportamento do animal. Há, portanto, uma lacuna entre os dispositivos atuais e um sistema verdadeiramente inteligente, capaz de perceber o ambiente, identificar o comportamento e reagir de forma automática e calibrada.

Diante desse cenário, o objetivo deste trabalho é demonstrar a viabilidade técnica de um sistema de adestramento automatizado para animais domésticos, baseado em visão computacional, capaz de unir percepção visual embarcada e resposta sonora adaptativa para auxiliar no controle de comportamentos indesejados. Para isso, propõe-se o desenvolvimento de um sistema que detecta a presença do animal em regiões do ambiente definidas como restritas, delimitadas pelo próprio usuário por meio de uma interface web hospedada no dispositivo, e emite um estímulo sonoro com parâmetros ajustáveis conforme a eficácia observada. O processamento das imagens e a tomada de decisão ocorrem de forma local, garantindo autonomia e baixa latência sem necessidade de conexão com a nuvem, integrando técnicas de detecção de objetos, verificação geométrica de interseção com regiões de interesse e controle parametrizado de frequências sonoras.

O trabalho está estruturado de modo a apresentar inicialmente a fundamentação teórica e o estado da arte sobre comportamento animal, visão computacional e sistemas de monitoramento. Em seguida, descreve-se o sistema desenvolvido e sua metodologia. Por fim, são apresentados e discutidos os resultados experimentais obtidos e as conclusões.

2 REVISÃO BIBLIOGRÁFICA E ESTADO DA ARTE

2.1 Comportamento Animal

O estudo do comportamento animal fornece a base conceitual para projetar sistemas de monitoramento e controle ambiental aplicados a animais domésticos. Na perspectiva etológica, o comportamento emerge da interação entre fatores fisiológicos, ambientais e de aprendizagem, de modo que a identificação de padrões e de contingências permite desenvolver intervenções sistemáticas, previsíveis e reprodutíveis (SKINNER, 1966).

Historicamente, dois marcos teóricos estruturam a compreensão moderna do aprendizado em animais: o condicionamento clássico e o condicionamento operante. Pavlov demonstrou que um estímulo inicialmente neutro pode eliciar uma resposta reflexa após emparelhamentos repetidos com um estímulo incondicionado — o *condicionamento clássico* (PAVLOV, 1927). Posteriormente, Skinner descreveu o *condicionamento operante* (SKINNER, 1953), segundo o qual a probabilidade de ocorrência de um comportamento varia em função de suas consequências; respostas seguidas de reforço tendem a aumentar de frequência, enquanto aquelas seguidas de consequências aversivas tendem a diminuir. Esses princípios fundamentam práticas contemporâneas de adestramento (LINDSAY, 2000) e possibilitam o desenho de sistemas automatizados que associam eventos ambientais a respostas programadas (por exemplo, emissão de sinal sonoro, jato de ar ou spray de água) como parte de uma política de reforço/controle (OVERALL, 2013).

Em ambientes domésticos, comportamentos exploratórios como subir em mesas, acessar lixeiras e jardins ou ingerir objetos/plantas são frequentes e podem representar riscos relevantes à saúde. A ingestão de plantas ornamentais potencialmente tóxicas, como espécies de Dieffenbachia, Lilium e Rhododendron, está associada a quadros de hipersalivação, vômito, alterações renais e outros sinais clínicos, a depender da espécie e da dose (ASPCA, 2025). O acesso a resíduos orgânicos e lixo aumenta a probabilidade de exposição a agentes bacterianos (e.g., Salmonella, E. coli) (GREENE, 2012) e a parasitas intestinais (BOWMAN, 2020), com potencial para gastroenterites e outras complicações. Tais fatores justificam o emprego de sistemas de vigilância e contenção que impeçam o animal de atingir zonas de risco (lixeira, cozinha, jardim, bancada), preferencialmente com intervenções de baixo impacto físico e alta previsibilidade, que não causem dor nem qualquer dano ao animal, assegurando que o estímulo tenha apenas caráter corretivo e não punitivo.

Sob a ótica do projeto de engenharia, a integração de técnicas de detecção automática (visão computacional e/ou sensores de presença) a fundamentos de análise do comportamento permite implementar controle em malha fechada. Ao identificar o comportamento-alvo, por exemplo, a aproximação da lixeira, aplica-se inicialmente um estímulo condicionado de baixa intensidade (sinal acústico breve) e, se necessário, procede-se a um escalonamento cuidadosamente parametrizado, como jato de ar de curta duração

(OVERALL, 2013). Com esse arranjo, compõe-se uma hierarquia de respostas que, junto a enriquecimento ambiental e reforço de comportamentos incompatíveis (LINDSAY, 2000), reduz a probabilidade futura do comportamento indesejado. Em termos de engenharia de controle, trata-se de um laço realimentado (OGATA, 2010) no qual sensores (detecção), controlador (lógica baseada no histórico/estado do animal) e atuadores (sinal acústico, jato de ar) fecham o ciclo sob critérios de estabilidade e desempenho definidos.

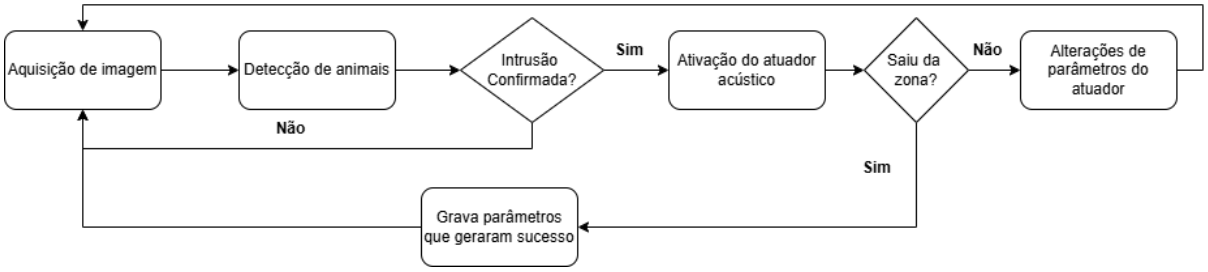


Figura 1: *Algoritmo descritivo do sistema: detecção → decisão de intrusão → acionamento acústico → verificação de saída → ajuste/loop. Fonte: Autor.*

Em síntese, os princípios de condicionamento clássico e operante, aliados a práticas contemporâneas de manejo comportamental, oferecem uma base sólida para a concepção de sistemas de monitoramento e intervenção automática que aumentem a segurança do ambiente sem sacrificar o bem-estar do animal.

2.2 Visão Computacional

A visão computacional compreende o conjunto de métodos e algoritmos que visam permitir que sistemas artificiais extraiam informação útil a partir de dados visuais, em geral imagens e vídeos (SZELISKI, 2022). Em termos gerais, o fluxo clássico envolve as etapas de aquisição e formação da imagem, pré-processamento e realce, segmentação e extração de características e, por fim, reconhecimento, detecção e/ou rastreamento de padrões (SZELISKI, 2022). Na etapa de aquisição e formação, modela-se a relação entre a cena, a óptica do sensor e a imagem resultante. Aspectos como exposição, ruído eletrônico, quantização e balanço de cores influenciam a qualidade da amostra; representações em diferentes espaços de cor (p. ex., RGB, HSV, YCbCr) são frequentemente empregadas para isolar componentes cromáticas ou de luminância, facilitando transformações subsequentes (GONZALEZ; WOODS, 2018).

O pré-processamento busca reduzir ruído e realçar estruturas relevantes. Técnicas lineares (filtros passa-baixa/alta) e não lineares (mediana, bilateral) são aplicadas para suavização e preservação de bordas; detectores de descontinuidades, como o Filtro Sobel (GONZALEZ; WOODS, 2018) e Filtro Canny (CANNY, 1986), e operações morfológicas (erodir, dilatar, abrir, fechar) auxiliam na evidência de contornos e na regularização de regiões (SZELISKI, 2022).

A segmentação consiste em particionar a imagem em regiões ou objetos de interesse. Métodos clássicos incluem limiarização global/adaptativa, como o método de Otsu (OTSU, 1979), crescimento de regiões, agrupamento (como k-means) e grafos. A escolha do método depende do contraste, da iluminação e da textura da cena (SZELISKI, 2022).

Uma vez segmentadas, as regiões são descritas por características (*features*) que compactam informação discriminativa: descritores locais (p. ex., SIFT (*Scale-Invariant Feature Transform*) (LOWE, 2004), HOG (*Histogram of Oriented Gradients*) (DALAL; TRIGGS, 2005)) capturam variações robustas de textura, gradiente e escala, enquanto características globais podem resumir forma e distribuição espacial.

Em problemas de reconhecimento e detecção, modelos classificadores mapeiam características para rótulos (classes), enquanto detectores estimam simultaneamente a presença e a localização de instâncias (tipicamente por caixas delimitadoras, *bounding boxes*). No contexto da visão computacional, em especial em tarefas de classificação de imagens e detecção de objetos, arquiteturas de aprendizado profundo (*deep learning*) tornaram-se predominantes: redes neurais convolucionais (*Convolutional Neural Networks*, CNNs) aprendem representações diretamente dos dados, superando abordagens de engenharia clássicas. Modelos como AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), VGG (SIMONYAN; ZISSERMAN, 2015) e ResNet (HE *et al.*, 2016) estabeleceram marcos em classificação. Em detecção, famílias baseadas em regiões, Faster R-CNN (REN *et al.*, 2015), e de estágio único, como YOLOv3 (REDMON; FARHADI, 2018), oferecem diferentes compromissos entre acurácia e velocidade. Para tarefas com dados limitados ou com restrições de processamento embarcado, estratégias como *transfer learning* e *fine-tuning* de modelos pré-treinados são amplamente adotadas (SZELISKI, 2022).

Além dos métodos teóricos, a área dispõe de um ecossistema consolidado de bibliotecas e frameworks que viabilizam a aplicação prática dessas técnicas. Entre as ferramentas de código aberto mais difundidas, destaca-se a *OpenCV* (*Open Source Computer Vision Library*) (OPENCV, 2025), que oferece implementações otimizadas para operações de filtragem, transformação geométrica, detecção de bordas e análise de movimento (BRADSKI, 2000). No domínio do aprendizado profundo, frameworks como *TensorFlow* (TENSORFLOW, 2025a), *PyTorch* (PYTORCH, 2025) e *ONNX Runtime* (MICROSOFT, 2025) fornecem suporte para o treinamento e a inferência de modelos CNN, com versões reduzidas (*TensorFlow Lite*, *PyTorch Mobile*) (TENSORFLOW, 2025b) voltadas ao processamento embarcado. Complementarmente, ferramentas como *MediaPipe* (EDGE, 2025) e *Ultralytics YOLO* (ULTRALYTICS, 2025) integram pipelines de detecção e rastreamento em tempo real, simplificando o uso de modelos pré-treinados em aplicações de visão computacional.

O rastreamento (*tracking*) em vídeo estende a detecção ao eixo temporal, associando medições quadro a quadro para estimar trajetórias. Filtros recursivos, como Kalman (linear) (KALMAN, 1960), Kalman Extendido (*Extended Kalman Filter*, EKF) (WELCH; BISHOP, 2001) e Kalman "sem cheiro" (*Unscented Kalman Filter*, UKF) (JULIER; UHLMANN, 2004), e métodos de associação de dados (p. ex., algoritmo Húngaro/*assignment* (KUHN, 1955; MUNKRES, 1957)) integram medidas ruidosas com modelos dinâmicos simples, enquanto abordagens modernas combinam detectores CNN com módulos de associação (*tracking-by-detection*) (SZELISKI, 2022). A avaliação de desempenho emprega métricas padronizadas, como acurácia (*accuracy*), precisão e revocação (*precision*/*recall*) (POWERS, 2011), precisões médias (*mean Average Precision*, mAP) e interseção sobre união (*Intersection over Union*, IoU) (EVERINGHAM *et al.*, 2010; LIN, T.-Y. *et al.*, 2014), o que permite comparar métodos de forma reprodutível.

Em síntese, a visão computacional oferece um arcabouço versátil para percepção visual, com um espectro que vai de técnicas clássicas, adequadas a cenários controlados e recursos limitados, a modelos de aprendizado profundo, capazes de generalização em ambientes complexos. A seleção das técnicas depende dos requisitos do problema (tempo real, consumo energético, precisão) e das características do cenário (iluminação, oclusões,

variabilidade intra/interclasse), critérios que serão retomados quando da definição dos componentes perceptuais do sistema proposto neste trabalho.

2.3 Detecção e Definição de Regiões de Interesse (ROI)

Em tarefas de visão computacional, a definição de *Regiões de Interesse* (ROI — *Regions of Interest*) é uma etapa essencial para concentrar o processamento e a análise em áreas específicas da imagem que contêm informação relevante (SZELISKI, 2022). O conceito de ROI está associado à ideia de restringir o domínio de processamento apenas a regiões de interesse, reduzindo a complexidade computacional e evitando o tratamento de áreas que não contribuem para a tarefa em questão (GONZALEZ; WOODS, 2018).

De maneira geral, uma ROI é definida geometricamente por um conjunto de vértices (x_i, y_i) que formam um polígono simples ou por um retângulo delimitador. O principal desafio consiste em determinar se um ponto, uma área ou um objeto identificado na imagem está total ou parcialmente contido nessa região (HORMANN; AGATHOS, 2001; O'ROURKE, 1998). Esse tipo de verificação é recorrente em sistemas de rastreamento, vigilância e inspeção industrial, nos quais o comportamento de um objeto dentro de uma zona específica é de interesse analítico.

Uma abordagem comum para essa verificação é o teste *ponto-em-polígono*, que consiste em determinar se um ponto de coordenadas (x, y), expressas no sistema de referência da imagem, pertence ao interior de um polígono. Entre os algoritmos mais utilizados estão o método da projeção de raios (*Ray-Casting*), também conhecido como regra par–ímpar, e o método do número de enrolamentos (*winding number*) (HORMANN; AGATHOS, 2001; O'ROURKE, 1998). Ambos possuem complexidade linear em relação ao número de vértices e são amplamente aplicados por sua eficiência e simplicidade de implementação (O'ROURKE, 1998).

Quando o objeto é representado por uma área, como uma caixa delimitadora (*bounding box*), a interseção entre essa área e a ROI pode ser calculada por meio de algoritmos de *polygon clipping*, como o de Sutherland–Hodgman para polígonos convexos (SUTHERLAND; HODGMAN, 1974). A partir dessa interseção, obtém-se a área comum $|A \cap B|$, permitindo definir medidas como o índice de sobreposição (*overlap ratio*) ou a interseção sobre união (*Intersection over Union* — IoU), amplamente empregadas para quantificar o grau de coincidência entre regiões (SZELISKI, 2022).

As técnicas de verificação espacial apresentadas são aplicáveis a uma ampla variedade de contextos: na robótica, para definir zonas de operação seguras, por exemplo *Speed and Separation Monitoring* (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2016) em células colaborativas (VILLANI *et al.*, 2018); em sistemas de vigilância, para detecção de intrusões e monitoramento perimetral (HU *et al.*, 2004); e em aplicações de análise de comportamento, para monitorar deslocamentos dentro de áreas delimitadas com rastreamento e/ou estimação de pose *markerless* (MATHIS *et al.*, 2018). O uso de ROIs, portanto, fornece uma estrutura matemática e computacional eficiente para interpretar relações espaciais entre objetos e regiões, sendo um componente fundamental de sistemas de visão que dependem de localização e contexto espacial (SZELISKI, 2022; GONZALEZ; WOODS, 2018).

2.4 Fontes Sonoras e Estímulos Condicionados

Os estímulos sonoros constituem um dos meios mais versáteis de interação entre sistemas automatizados e organismos vivos (OVERALL, 2013). Na etologia e na psicologia comportamental, o som é amplamente utilizado como reforço, sinal discriminativo ou estímulo condicionado, capaz de evocar respostas comportamentais específicas conforme sua frequência, intensidade e padrão temporal (SKINNER, 1938). Quando empregados de forma controlada, os estímulos auditivos possibilitam a criação de associações previsíveis entre eventos ambientais e ações desejadas, princípio fundamental para o desenvolvimento de sistemas de condicionamento e treinamento automatizados (SKINNER, 1953).

A percepção sonora em animais domésticos, especialmente cães e gatos, apresenta ampla faixa de sensibilidade: enquanto o ouvido humano percebe, em média, frequências entre 20 Hz e 20 kHz, cães e gatos estendem essa faixa para aproximadamente 45 kHz e 85 kHz, respectivamente (HEFFNER, 1983; HEFFNER; HEFFNER, 1985). Essa diferença justifica o uso de fontes sonoras de alta frequência, imperceptíveis aos humanos, mas eficazes como estímulo aversivo ou sinal de alerta em dispositivos eletrônicos. Para fins de segurança e bem-estar, contudo, recomenda-se operar em bandas ultrassônicas inaudíveis aos humanos (aproximadamente 20 a 35 kHz) (FLETCHER *et al.*, 2018), mantendo o nível de pressão sonora (*Sound Pressure Level*, SPL) na posição do animal, de forma conservadora, em $\leq$ 70–75 dB no terço de oitava centrado em 20 kHz (limites de exposição pública para humanos adotados aqui como teto prudencial) (INTERNATIONAL COMMISSION ON NON-IONIZING RADIATION PROTECTION, 2024; HEALTH CANADA, 1991) e privilegiando pulsos breves com baixa taxa de repetição (FLETCHER *et al.*, 2018). Em cães, há evidência de que o limiar de dor auditiva ocorre em torno de aproximadamente 95 dB (FAGUNDES *et al.*, 2018), e exposições prolongadas em canis na faixa de 100–108 dB associam-se a alterações mensuráveis de audição (SCHEIFELE *et al.*, 2012), o que reforça a necessidade de margens amplas e de evitar tons contínuos. Assim, a seleção de frequência e a parametrização de intensidade/duração devem buscar eficácia comportamental sem desconforto, mantendo níveis abaixo desses patamares e com monitoramento do efeito no indivíduo.

Em sistemas automatizados, as fontes sonoras podem assumir múltiplas funções: (i) indicar eventos internos, como o acionamento de atuadores ou a detecção de um comportamento; (ii) servir como reforço condicionado, associado a consequências positivas ou negativas; e (iii) atuar como sinal discriminativo, delimitando contextos de ação ou inibição (LINDSAY, 2000). A escolha da frequência, duração e intensidade do som deve considerar o limiar auditivo e a sensibilidade da espécie, de modo a evitar respostas de medo, estresse ou habituação precoce.

O processo de condicionamento sonoro baseia-se nos mesmos princípios formulados por Pavlov e Skinner: um estímulo auditivo neutro (como um breve tom) é emparelhado a um estímulo incondicionado (por exemplo, um jato de ar ou spray de água), de modo que, após repetições, o som sozinho passa a eliciar a resposta comportamental desejada (PAVLOV, 1927). Essa associação pavloviana pode ser reforçada por mecanismos de condicionamento operante, nos quais o sistema automatizado detecta o comportamento e aplica o estímulo auditivo como consequência imediata, ajustando a frequência ou intensidade conforme a reincidência da ação.

A literatura evidencia que respostas condicionadas auditivas são mais eficazes quando há uma clara relação temporal entre o som e o evento consequente (MACKINTOSH, 1974). Assim, em sistemas automatizados de controle comportamental, a latência entre a detecção do comportamento e a emissão do som deve ser minimizada para preservar a

associação. Implementações baseadas em microcontroladores permitem que esse tempo de resposta seja da ordem de milissegundos, o que torna viável o condicionamento preciso em ambientes domésticos (ZAGITOV *et al.*, 2024; SANTOS; COELHO; OLIVEIRA, 2024).

Por fim, o uso de estímulos sonoros em sistemas inteligentes exige uma abordagem ética e segura. Estudos mostram que sons intensos ou prolongados podem causar estresse e alterações fisiológicas em animais (BEERDA *et al.*, 1998), de modo que a utilização de estímulos auditivos deve priorizar intensidades moderadas, curtas e de baixa frequência de repetição. A adoção de estratégias de reforço negativo leve e preditivo, combinadas a alternativas de reforço positivo (como cessar o som após o afastamento), favorece o aprendizado e reduz o desconforto, alinhando a automação comportamental aos princípios de bem-estar animal.

2.5 Sistemas de Monitoramento Automatizado

Os sistemas de monitoramentos automatizados combinam tecnologias de sensoriamento, processamento embarcado e comunicação de dados para observar e registrar eventos de forma contínua, com mínima intervenção humana. A popularização da Internet das Coisas (*Internet of Things*, IoT) (ASHTON, 2009) e dos microcontroladores de baixo custo possibilitou o desenvolvimento de arquiteturas distribuídas, nas quais múltiplos dispositivos atuam de maneira integrada para coleta e análise de informações ambientais (ATZORI; IERA; MORABITO, 2010). Em contextos domésticos, esses sistemas assumem funções de vigilância, automação residencial, controle de acesso, monitoramento de saúde e, mais recentemente, acompanhamento do comportamento de animais de estimação.

De modo geral, um sistema de monitoramento automatizado é composto por quatro camadas funcionais: (i) aquisição de dados, (ii) processamento local, (iii) comunicação e (iv) atuação ou armazenamento (AL-FUQAHA *et al.*, 2015). A camada de aquisição compreende sensores ópticos (por exemplo, módulos ToF como o VL53L0X), sensores de movimento (sensores infravermelhos passivos, acelerômetros) e sensores de massa/contato (células de carga lidas via conversor de instrumentação), responsáveis pela coleta de informações do ambiente físico (STM MICROELECTRONICS, 2024; AVIA SEMICONDUCTOR, 2012). O processamento local, executado em microcontroladores ou sistemas embarcados, como ESP32 (ESPRESSIF SYSTEMS, 2020), Raspberry Pi 4 (RASPBERRY PI LTD., 2025) ou NVIDIA Jetson Orin Nano (NVIDIA, 2025), realiza a filtragem, análise e decisão sobre os dados recebidos, podendo empregar técnicas de visão computacional ou aprendizado de máquina em tempo real (SHI *et al.*, 2018). A camada de atuação pode envolver motores de passo, enquanto o armazenamento/temporalidade pode recorrer a RTCs, e a infraestrutura de energia pode incluir reguladores buck. A comunicação entre os módulos ocorre por meio de protocolos cabeados (UART, I2C, SPI, Ethernet) ou sem fio (Wi-Fi, Bluetooth, ZigBee, LoRa), dependendo das restrições de energia, alcance e largura de banda. Em sistemas IoT, o modelo de comunicação publicação/assinatura (*publish-subscribe*), viabilizado por protocolos como Transporte de Telemetria para Enfileiramento de Mensagens (*Message Queuing Telemetry Transport*, MQTT) é amplamente utilizado por permitir a troca eficiente de dados entre sensores, atuadores e servidores (AL-FUQAHA *et al.*, 2015). Essa conectividade possibilita que dispositivos periféricos transmitam dados para plataformas na nuvem, onde são realizados armazenamento, análise estatística e visualização.

A camada final, de atuação, compreende os dispositivos responsáveis por reagir

às condições detectadas: emissores sonoros, atuadores mecânicos, LEDs (*Light-Emitting Diode*) de sinalização ou comandos remotos. Quando integradas em um ciclo de controle, essas ações podem ocorrer de forma autônoma, baseadas em regras lógicas ou em modelos de decisão probabilística. Sistemas desse tipo operam em malha fechada, reagindo de forma adaptativa a eventos do ambiente, por exemplo, detectar um comportamento indesejado e emitir um som corretivo, conforme descrito neste trabalho.

No que se refere à arquitetura de software, as abordagens mais comuns envolvem o uso de plataformas embarcadas com suporte a multitarefas e comunicação em rede, como o FreeRTOS (FREERTOS, 2025) e o MicroPython (MICROPYTHON, 2025), além do uso de bibliotecas de visão computacional como OpenCV (OPENCV, 2025) e TensorFlow Lite (TENSORFLOW, 2025b). Ferramentas como o FreeRTOS permitem a execução simultânea de tarefas de aquisição, inferência e comunicação, otimizando o uso de recursos e reduzindo a latência entre detecção e resposta (SHI *et al.*, 2018). Quando aplicados a sistemas de controle comportamental, tais recursos permitem o monitoramento contínuo de ambientes, o registro de eventos e o acionamento automático de mecanismos de correção com base em critérios previamente definidos.

Por fim, a confiabilidade e a segurança desses sistemas dependem de fatores como estabilidade de rede, integridade dos sensores e proteção dos dados transmitidos. Questões como perda de pacotes, interferência eletromagnética, ruído nos sinais de sensores e falhas de calibração podem comprometer a precisão das detecções e, consequentemente, a eficácia da resposta automatizada. Assim, o projeto de sistemas de monitoramento deve buscar um equilíbrio entre desempenho, custo e robustez, empregando estratégias de redundância e calibração periódica para garantir resultados consistentes e replicáveis.

2.6 Processamento em tempo de execução em sistemas de visão embarcados

Soluções de monitoramento baseadas em visão computacional para ambiente doméstico precisam operar em tempo real, ou quase, para que o estímulo aplicado ainda esteja causalmente associado ao comportamento do animal. Na prática, isso significa fechar o ciclo aquisição de imagem $\rightarrow$ detecção $\rightarrow$ decisão $\rightarrow$ atuação dentro de uma janela de alguns centésimos de segundo, mantendo ao mesmo tempo consumo de energia compatível com um sistema embarcado.

A literatura de visão embarcada mostra que esse compromisso entre latência, taxa de quadros e consumo de potência é um ponto central de projeto. Revisões recentes de sistemas de visão com aprendizado profundo embarcado descrevem arquiteturas que vão de plataformas flexíveis com dezenas de watts de consumo até soluções ultraotimizadas que operam abaixo de 1 W, organizadas em categorias de sistemas *near-sensor*, processamento em chip adjacente ao sensor, e *in-sensor*, processamento dentro do próprio chip do sensor, para reduzir movimentação de dados e atrasos de processamento (FABRE *et al.*, 2024). Estudos de detecção de objetos em dispositivos de baixo consumo mostram que a escolha do modelo (por exemplo, variações pequenas de redes da família YOLO ou arquiteturas leves como MobileNet e EfficientDet) influencia diretamente o compromisso entre acurácia, taxa de quadros e utilização de CPU ou GPU, com recomendações específicas de modelos para diferentes faixas de desempenho em plataformas como Raspberry Pi e Jetson Nano (ZAGITOV *et al.*, 2024; SANTOS; COELHO; OLIVEIRA, 2024). Esses resultados reforçam que a seleção do algoritmo de visão não pode ser feita apenas pela acurácia em banco de

testes: é necessário validá-lo no hardware alvo, com restrições de tempo de resposta bem definidas.

Nesse contexto, o campo de Tiny Machine Learning (TinyML) busca levar redes neurais compactas para microcontroladores e outros dispositivos com poucos quilobytes de memória, privilegiando inferência local com baixíssimo consumo de energia (SCHIZAS *et al.*, 2022; LIN, J. *et al.*, 2023). Revisões sistemáticas apontam que o uso de técnicas como quantização, poda e busca de arquiteturas neurais orientada a recursos permite executar classificação de imagens, detecção de eventos e outras tarefas de visão em dispositivos na faixa de miliwatts, desde que o algoritmo e a pilha de sistema sejam projetados em conjunto (LIN, J. *et al.*, 2023). Ferramentas como o TensorFlow Lite para microcontroladores fornecem um ambiente de execução cujo núcleo ocupa cerca de 16 kB em núcleos ARM Cortex-M e dispensa sistema operacional ou alocação dinâmica de memória, viabilizando a implantação dessas redes em plataformas restritas (TENSORFLOW, 2021).

Para o sistema proposto neste trabalho, essas referências sustentam a necessidade de especificar claramente os requisitos de latência do laço de controle, desde a captura da imagem até o acionamento acústico, de escolher algoritmos de visão e hardware que alcancem essa latência dentro do orçamento de energia disponível.

3 DESENVOLVIMENTO DO PROJETO

Antes de iniciar, é importante dar um panorama geral ao leitor do que será elaborado. De maneira geral, a Figura 2 apresenta a arquitetura geral do sistema, mapeando o fluxo de sinais e a conexão entre os blocos. O sistema estrutura-se a partir de um nó Coletor, que realiza a interface direta com o ambiente por meio de uma câmera e de um atuador sonoro. Os dados de imagem são direcionados a um nó processador para a execução dos algoritmos de detecção e tomada de decisão. Posteriormente, o status do monitoramento é enviado a uma plataforma web, permitindo o acompanhamento remoto por parte do usuário.

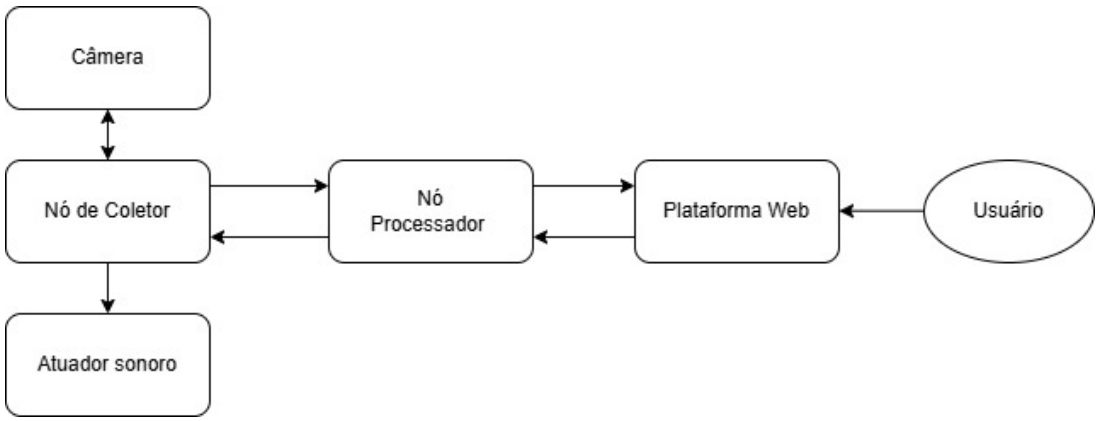


Figura 2: *Proposta do diagrama de blocos da topologia distribuída do sistema: fluxo de dados bidirecional entre o nó coletor, o nó processador e a interface web.*

3.1 Desenvolvimento da Rede Neural de Detecção

Esta subseção descreve o processo de seleção da arquitetura de aprendizado profundo (*deep learning*), a composição e estruturação do conjunto de dados (*dataset*), os métodos de aumento de dados (*data augmentation*) aplicados para robustez ambiental, bem como a estratégia de treinamento e validação do modelo computacional.

3.1.1 Seleção da Arquitetura do Modelo

Para viabilizar a execução de algoritmos de visão computacional em sistemas embarcados com recursos computacionais limitados, como microcontroladores e minicomputadores de placa única, faz-se necessária a adoção de arquiteturas otimizadas. Neste

projeto, optou-se pela utilização da rede **YOLOv8** em sua variante **Nano (YOLOv8n)**, desenvolvida pela Ultralytics (JOCHER; CHAURASIA; QIU, 2023a).

A escolha da arquitetura justifica-se pelo equilíbrio entre custo computacional e acurácia de detecção. Diferente de modelos convolucionais tradicionais baseados em abordagens de duas etapas (*two-stage detectors*, como Faster R-CNN) (REN *et al.*, 2015), a família YOLO (*You Only Look Once*) formula a detecção de objetos como um problema de regressão único, mapeando pixels diretamente para coordenadas de caixas delimitadoras (*bounding boxes*) e probabilidades de classe em uma única passagem pela rede (REDMON; DIVVALA *et al.*, 2016). A versão Nano minimiza o número de parâmetros ativos e operações de ponto flutuante, tornando-a ideal para a implementação em cenários de inferência em tempo real na borda (*edge computing*) (SHAFIQUE MUHAMMAD, 2021).

Para mitigar a necessidade de volumes massivos de dados iniciais e acelerar a convergência do aprendizado, empregou-se a técnica de Aprendizado por Transferência (*Transfer Learning*) (PAN; YANG, 2009). O modelo foi inicializado utilizando os pesos sinápticos pré-treinados no *dataset* de ampla escala *Common Objects in Context* (COCO) (LIN, T.-Y. *et al.*, 2014), benchmark consolidado na literatura de visão computacional, composto por mais de 330 mil imagens anotadas abrangendo 80 categorias de objetos do cotidiano, amplamente utilizado para pré-treinamento de modelos de detecção e segmentação. A partir dessa base de extratores de características (*features*) visuais primitivas já consolidadas, tais como bordas, texturas e formas geométricas, realizou-se o ajuste fino (*fine-tuning*) das camadas finais para especializar o modelo na identificação exclusiva de duas classes de interesse: cães (*dogs*) e gatos (*cats*).

3.1.2 Composição e Estruturação do Dataset

O conjunto de dados utilizado para o treinamento foi concebido a partir da fusão e normalização de fontes públicas distintas e consolidadas na literatura de visão computacional. Para a amostragem das classes de interesse, as instâncias associadas aos canídeos foram extraídas do *Stanford Dogs Dataset* (KHOSLA *et al.*, 2011), base de dados amplamente reconhecida contendo imagens de diversas raças com anotações complexas. Por sua vez, o conjunto de amostras voltado à classe dos felinos foi obtido a partir do repositório especializado *images.cv*, através do pacote *Cat Image Classification Dataset* (IMAGES.CV, 2026)[1]. O *dataset* bruto original continha anotações em formatos heterogêneos, incluindo arquivos estruturados em *Extensible Markup Language* (XML) para cães e arquivos de texto plano (*.txt*) para gatos.

Como etapa de pré-processamento, desenvolveu-se um *pipeline* de normalização automatizado para converter todas as coordenadas de caixas delimitadoras para o padrão restrito exigido pela arquitetura YOLO. Neste formato, cada imagem possui um arquivo correspondente *.txt* onde cada linha representa uma instância de objeto codificada por cinco valores flutuantes normalizados entre 0 e 1:

$$\text{Classe} \quad X_{\text{centro}} \quad Y_{\text{centro}} \quad \text{Largura} \quad \text{Altura} \tag{1}$$

[1] O repositório *images.cv* é uma plataforma comercial de conjuntos de dados rotulados para visão computacional. Embora não constitua uma publicação acadêmica convencional, o dataset utilizado encontra-se publicamente acessível e documentado, sendo referenciado pela URL de acesso e data de consulta, conforme prática adotada para fontes digitais nas normas ABNT.

As instâncias associadas à classe de cães foram mapeadas uniformemente sob o identificador numérico `0`, ao passo que os espécimes da classe de gatos foram atribuídos ao identificador `1`.

Para contextualizar o modelo ao ambiente de aplicação final e reduzir a ocorrência de falsos positivos induzidos por elementos estáticos da residência, integrou-se ao *pipeline* um conjunto secundário de imagens de plano de fundo denominado room-dataset. Este subconjunto foi estruturado a partir da base de dados pública *House Rooms*, hospedada na plataforma Kaggle (KAGGLE, 2024). O repositório compreende imagens categorizadas de ambientes internos domésticos reais, tais como banheiros (*bathroom*), quartos (*bedroom*), salas de estar (*living*), cozinhas (*kitchen*) e escritórios (*office*).

A composição proporcional do *dataset* final ($42,5\%$ para a classe 0, $42,5\%$ para a classe 1 e 15% de imagens de falso plano de fundo) foi delimitada com base em critérios de regularização estatística para redes neurais convolucionais profundas. O percentual de 15% de amostras de *background* puras (room-dataset) foi adotado para reduzir o índice de Falsos Positivos por Imagem (*False Positives Per Image* – FPPI). Em ambientes domésticos confinados, o plano de fundo tende a apresentar alta correlação espacial continua; a inserção dessa proporção de imagens vazias força a função de perda (*loss function*) da YOLOv8n a penalizar ativações espúrias em estruturas estáticas como sofás, tapetes e sombras, garantindo o balanceamento numérico ideal entre os gradientes das classes ativas e o gradiente nulo do extrato de fundo. Assim, 6.250 amostras foram utilizadas, distribuídas conforme a Tabela 1.

Tabela 1: *Distribuição quantitativa e proporcional das amostras do dataset final.*

Categoria de Amostra	ID da Classe	Proporção (%)	Quantidade Absoluta
Instâncias de Cães	0	42,5%	2.656
Instâncias de Gatos	1	42,5%	2.656
Cenários Puros	N/A	15,0%	938
Total Geral	**-**	**100,0%**	**6.250**

Após a consolidação das imagens de plano de fundo e das classes extraídas, o *pipeline* realiza o processo de colagem contextual utilizando técnicas de segmentação automática para remoção de fundo. Esse procedimento isola o objeto de interesse (animal) eliminando texturas espúrias de sua origem e permitindo sua inserção direta sobre cenários residenciais aleatórios. O resultado prático de uma amostra sintética final gerada para o treinamento do modelo é apresentado na Figura 3.

A inclusão de 15% de imagens de cenários puros (completamente desprovidas de rótulos de objetos) constitui uma técnica regulatória crucial recomendada para o treinamento de modelos YOLO (JOCHER; CHAURASIA; QIU, 2023a). Essa estratégia força o algoritmo de treinamento a penalizar o gradiente quando caixas delimitadoras são previstas erroneamente sobre áreas vazias do ambiente doméstico, forçando a rede a aprender de forma explícita a diferença entre o plano de fundo inercial e a presença real de um animal, reduzindo a incidência de alarmes falsos (JOCHER; CHAURASIA; QIU, 2023b).

3.1.3 Técnicas de Aumento de Dados (Data Augmentation) e Sobreposição Contextual

Visando expandir artificialmente a variabilidade do *dataset* e aproximar as amostras das condições reais de captura, frequentemente sujeitas a artefatos de movimento, variações

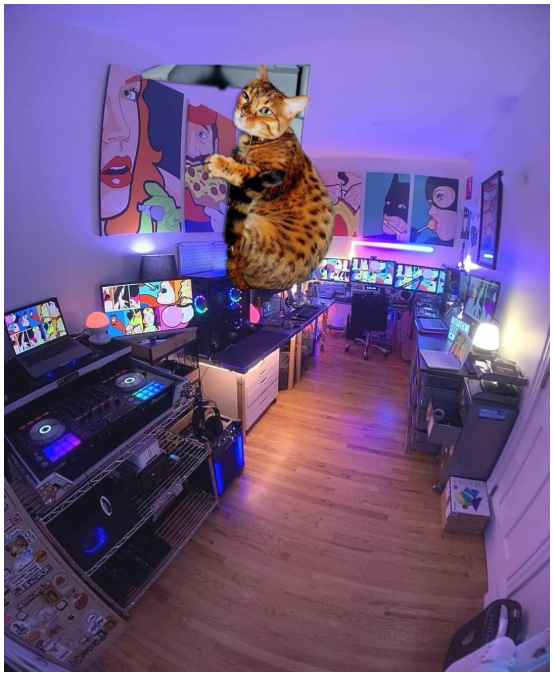


Figura 3: *Exemplo de amostra sintética gerada pelo pipeline de colagem contextual através de remoção de fundo e sobreposição, integrando o objeto de interesse ao room-dataset.Fonte: Autor*

de luminosidade e oclusões parciais, implementou-se um *pipeline* matemático estocástico de aumento de dados (*data augmentation*) operando em tempo de execução, técnica fundamental para evitar o *overfitting* em redes convolucionais profundas (SHORTEN; KHOSHGOFTAAR, 2019). O overfitting ocorre quando o modelo aprende a memorizar os exemplos de treinamento em vez de generalizar os padrões subjacentes, resultando em desempenho elevado sobre o conjunto de treino mas degradado em dados novos, comportamento crítico neste contexto, onde o modelo deve reconhecer animais em ambientes domésticos reais distintos dos cenários sintéticos utilizados no treinamento.

Para cada imagem contendo um animal processado, aplicou-se sequencialmente o seguinte conjunto de transformações lineares e não lineares:

1. **Transformação Geométrica Afim (Rotação)**: Aplicação de uma matriz de rotação bidimensional com ângulo θ selecionado aleatoriamente dentro do intervalo de $[-15°, +15°]$. A operação é definida pela Equação (2):

$$M = \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \tag{2}$$

   Onde $\alpha = \cos(\theta)$, $\beta = \sin(\theta)$ e o centro de rotação é fixado no centro geométrico da imagem. Para preencher os vazios gerados nos cantos das imagens rotacionadas, adotou-se o método de extrapolação por reflexão de borda (*border reflection*), preservando a continuidade das texturas para a rede convolucional.

2. **Perturbação de Iluminação via Look-Up Table (LUT)**: Ajuste não linear de brilho por meio da multiplicação escalar dos canais de cor por um fator aleatório $\gamma \in [0,75, 1,25]$. Os valores resultantes são truncados dinamicamente no intervalo de intensidade de pixel $[0, 255]$ utilizando a operação matemática de *clip*, conforme expresso na Equação (3):

$$f(x) = \min(\max(x \cdot \gamma, 0), 255) \tag{3}$$

Esta técnica simula variações de luz natural e sombras comuns ao longo do dia nos cômodos residenciais.

3. **Suavização Espacial (Desfoque Gaussiano)**: Sob probabilidade estrita de 50%, a imagem é submetida a um filtro convolucional utilizando um núcleo (*kernel*) Gaussiano de dimensões 3×3. A operação visa emular imperfeições de foco causadas pela movimentação rápida do animal (*motion blur*) ou limitações ópticas do sensor de captura.

Após a aplicação do bloco de modificações isoladas, executou-se a etapa de Sobreposição Contextual via Matriz de Máscara Binária, inspirada em pipelines avançados de *Cut-and-Paste* para detecção de objetos (DWIBEDI; MISRA; HEBERT, 2017). O algoritmo extrai uma máscara binária $M(x)$ do animal de interesse, onde pixels pertencentes ao animal assumem valor 1 e o fundo original assume 0. A remoção do fundo original é uma etapa crítica do *pipeline*: ao isolar o animal de sua cena de origem, evita-se que a rede neural aprenda associações espúrias entre as características visuais do animal e os padrões de textura, cor ou iluminação presentes no cenário de origem do dataset (e.g., grama, chão de madeira, paredes específicas). Sem essa etapa, o modelo poderia aprender a detectar o animal condicionado ao fundo original, resultando em falsos negativos em ambientes domésticos onde esses contextos não estão presentes. Ao recompor o animal sobre cenários residenciais aleatórios do room-dataset, força-se a rede a aprender exclusivamente as características morfológicas do animal, independentemente do ambiente ao redor.

A imagem sintética final $G(x)$ é gerada a partir da substituição condicional de pixels sob a resolução de 320×320 pixels, mapeando o objeto segmentado diretamente sobre uma imagem de ambiente aleatória oriunda do room-dataset, conforme a Equação (4):

$$G(x) = M(x) \cdot I_{\text{animal}}(x) + \Big(1 - M(x)\Big) \cdot I_{\text{cenario}}(x) \tag{4}$$

Esse procedimento de remoção automatizada e sobreposição preserva as bordas e a opacidade real do animal sobre o novo plano de fundo residencial. Como consequência, força-se as camadas convolucionais profundas da rede a desacoplarem as características intrínsecas e gradientes morfológicos do animal em relação às texturas e formas do cenário de fundo doméstico, garantindo maior robustez à arquitetura.

3.1.4 Estratégia de Treinamento e Validação Prática

Para garantir a capacidade de generalização do modelo e a correta validação estatística das métricas de desempenho, o volume totalizado de 6.250 amostras geradas foi particionado de maneira pseudoaleatória em dois conjuntos disjuntos através de amostragem estratificada, adotando-se a proporção de 80% para treinamento (5.000 imagens) e 20% para validação (1.250 imagens). As amostras de validação funcionam como um conjunto de teste cego contínuo, permitindo mensurar a perda (*loss*) em dados nunca vistos pela rede ao término de cada época (JAMES *et al.*, 2013).

O processo de otimização dos pesos sinápticos foi configurado com hiperparâmetros restritos para evitar a divergência dos gradientes devido ao uso de aprendizado por transferência. Fixou-se o limite nominal do processo em 250 épocas e adotou-se o tamanho de lote (*batch size*) de 16 unidades. O comportamento dinâmico do aprendizado foi controlado pelas seguintes diretrizes:

- **Taxa de aprendizado Inicial Reduzida ($lr0$)**: Fixada em $0,001$ (um fator dez vezes menor que o padrão nativo da arquitetura). A redução do passo do gradiente impede a destruição abrupta dos extratores de características complexas previamente consolidados na inicialização dos pesos do COCO (YOSINSKI *et al.*, 2014).
- **Período de aquecimento (*Warmup Epochs*)**: Configuração de um estágio inicial de 3 épocas onde a taxa de aprendizado cresce linearmente a partir de valores próximos a zero até atingir o patamar nominal de $0,001$. Esse procedimento estabiliza as flutuações iniciais dos gradientes decorrentes da introdução do ruído gerado pelas técnicas de *blending* (GOYAL *et al.*, 2017).
- **Mecanismo de parada antecipada (*Early Stopping*)**: Implementou-se o monitoramento contínuo da métrica de precisão média ponderada *mean Average Precision* no limiar de interseção sobre união de 0,50 a 0,95 (mAP_{50-95}). Definiu-se um parâmetro de paciência de 20 épocas ($patience = 20$). Caso o modelo falhe em apresentar ganho incremental na métrica de validação por 20 iterações consecutivas sobre o ciclo completo de dados, o processo é interrompido defensivamente (PRECHELT, 1998).

3.1.5 Topologia Distribuída e Análise Comparativa de Hardware

A engenharia de sistemas embarcados voltados à visão computacional exige o dimensionamento correto do compromisso entre o balanço energético, custo e a capacidade computacional (SHAFIQUE MUHAMMAD, 2021). Neste projeto, o isolamento físico entre o ponto de captura visual e o processamento da rede neural motivou a divisão do sistema em dois nós principais: um módulo coletor periférico de aquisição e um nó processador central de inferência, conforme apresentado no diagrama de blocos da Figura 4.

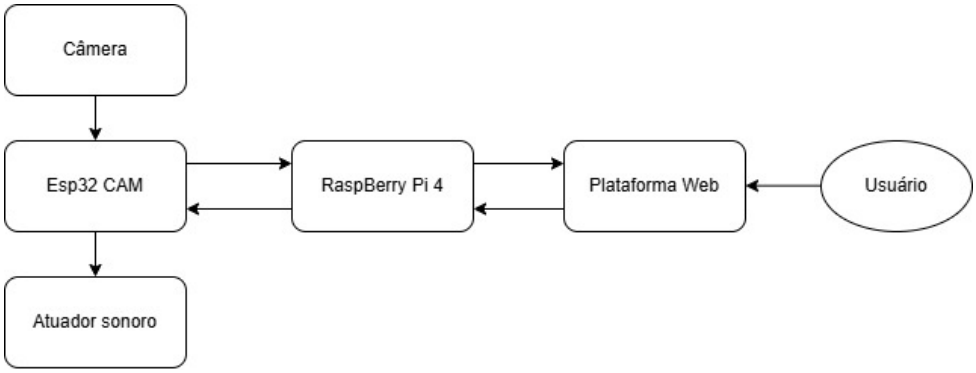


Figura 4: *Diagrama de blocos da topologia distribuída do sistema: fluxo de dados bidirecional entre o ESP32, a Raspberry Pi 4 e a interface web.*

A seguir, apresenta-se uma análise comparativa multifatorial das alternativas de hardware avaliadas, segregada por classe de dispositivo e função de barramento.

3.1.5.1 Análise do Nó Periférico de Aquisição: ESP32-CAM e Alternativas Microcontroladas

O nó periférico requer um dispositivo com dimensões reduzidas, baixo consumo energético e versatilidade de periféricos para digitalização de imagem e controle básico de

Tabela 2: *Análise comparativa de alternativas para o nó periférico de aquisição.*

Parâmetro Técnico	Arduino Uno + Shield Câmera	ESP32-CAM
Arquitetura	ATmega328P (8-bit)	Tensilica LX6 (32-bit Dual-Core)
Frequência de Clock	16 MHz	240 MHz
Memória SRAM Nativa	2 KB	520 KB
Extensão PSRAM	Não suportado	4 MB
Conectividade Sem Fio	Nenhuma (Exige Shield)	Wi-Fi 802.11 b/g/n e Bluetooth
Custo Médio de Mercado (maio, 2026)	R$ 90,00 - R$ 120,00	R$ 45,00 - R$ 60,00

atuadores. A Tabela 2 estabelece o contraste técnico entre o circuito integrado selecionado (ESP32-CAM) e outra plataforma microcontrolada de ampla difusão no meio acadêmico e de prototipagem, o Arduino Uno (pareado a um *shield* de câmera externo).

A limitação das arquiteturas de 8 bits para esta aplicação fundamenta-se na restrição de alocação de buffers de vídeo. Um único quadro digitalizado sob a resolução QVGA (320×240 pixels) em formato bruto ocupa um espaço de memória volátil que excede a capacidade total de memória SRAM de um Arduino Uno.

A ESP32-CAM destaca-se ao integrar barramentos de memória pseudoestática (PSRAM) de 4 MB, operando a 240 MHz, o que viabiliza o streaming de vídeo de forma nativa. Adicionalmente, o dispositivo apresenta vantagens em termos de portabilidade, registrando cerca de um quarto da massa e metade das dimensões da solução modular concorrente. A sua versatilidade se traduz na presença de transceptores de rádio integrados (Wi-Fi) e temporizadores de hardware capazes de gerar modulação por largura de pulso (PWM), permitindo que o dispositivo atue como um nó inteligente sem fio autônomo por um custo financeiro reduzido.

3.1.5.2 Análise do Nó Processador Central: Raspberry Pi 4 e Alternativas Single-Board

O nó processador atua como o servidor local de borda, encarregado de decodificar o fluxo M-JPEG e computar os tensores convolucionais da rede neural. A Tabela 3 detalha a comparação entre o computador de placa única escolhido (Raspberry Pi 4) e a plataforma de hardware focado em aceleração de inteligência artificial NVIDIA Jetson Nano.

Tabela 3: *Análise comparativa de alternativas para o nó central de inferência.*

Métrica	NVIDIA Jetson Nano	Raspberry Pi 4
Processamento	GPU Maxwell - 128 núcleos	CPU ARM - 4 núcleos
Memória RAM	4 Gb	4Gb
Desempenho YOLOv8n	$\sim$ 25 - 30 FPS	$\sim$ 3 - 5 FPS
Consumo de Potência Máximo	10 W a 15 W	5 W a 7 W
Massa Aproximada	$\sim$ 140 g (com dissipador)	$\sim$ 46 g
Custo Médio de Mercado (maio, 2026)	R$ 1.200,00 - R$ 1.500,00	**R$ 450,00 - R$ 600,00**

A definição das especificações deste projeto estabeleceu uma meta de desempenho de 5 FPS, com margem de tolerância para oscilações inerentes à capacidade de processamento da plataforma de hardware utilizada. Estudos de cronobiologia e monitoramento comportamental por acelerometria e visão em ambientes residenciais demonstram que o padrão de atividade locomotora e os ritmos diários de deslocamento de animais domésticos (*Felis catus* e *Canis familiaris*) são predominantemente compostos por movimentos de baixa intensidade e repouso (PICCIONE *et al.*, 2013; MARTÍNEZ-BYER *et al.*, 2023).

Contudo, esses períodos são intercalados por surtos dinâmicos de deslocamento; análises cinemáticas e de marcha indicam que o trote e a caminhada rápida dessas espécies cobrem bandas de velocidade tipicamente situadas entre 1,0 m/s e 2,5 m/s (VOSS *et al.*, 2020; SCHILLING; CARRIER, 2008; BELLARDITA; KIEHN, 2015). Sob a ótica da teoria de controle e amostragem de sinais, uma taxa de amostragem fixada em 5 Hz estabelece um período discreto de amostragem de 200 ms. Este intervalo limita o deslocamento espacial máximo do elemento dinâmico a menos de 0,5 m entre quadros sucessivos, mesmo em cenários de marcha heterogênea (VOSS *et al.*, 2020). Portanto, a frequência adotada mostra-se plenamente suficiente para mapear o posicionamento e disparar as ações de controle acústico de forma rápida e preditiva, sem perdas por latência cumulativa no laço fechado. Sob a ótica desta meta de 5 FPS, o emprego da plataforma NVIDIA Jetson Nano caracterizaria um caso de superdimensionamento de hardware (SHAFIQUE MUHAMMAD, 2021). Embora a Jetson Nano entregue taxas elevadas de quadros devido à sua GPU integrada, ela apresenta um custo de mercado significativamente superior ao da Raspberry Pi 4 e encontra-se oficialmente descontinuada pelo fabricante, limitando o suporte de longo prazo a bibliotecas modernas de *deep learning*.

Sob o ponto de vista mecânico e físico, a Raspberry Pi 4 apresenta-se como uma solução mais compacta e leve, registrando cerca de um terço do peso da plataforma concorrente por dispensar blocos massivos de dissipação térmica passiva. A Raspberry Pi 4 consolida sua viabilidade ao aliar custo acessível, amplo ciclo de vida de mercado e consumo de pico limitado a 7 W, enquanto suas extensões vetoriais NEON processam a YOLOv8n na faixa de 3 a 5 FPS (VIVEKANANDA *et al.*, 2025), superando a meta de engenharia estipulada com eficiência energética.

3.1.6 Protocolo de Comunicação e Arquitetura de Rede

O transporte do fluxo de vídeo estruturado entre o módulo coletor e o processador embarcado utiliza a pilha de protocolos de rede TCP/IP sobre uma rede local sem fio sob o padrão IEEE 802.11 b/g/n (Wi-Fi).

O modelo de arquitetura de rede adota a topologia Cliente-Servidor com fluxo bidirecional de dados, estruturada sob a seguinte dinâmica de barramento de software:

1. O microcontrolador ESP32-CAM atua como um servidor HTTP leve (*Micro Web Server*) persistente em uma porta de rede dedicada.

2. Ao inicializar, o firmware estabelece o mapeamento do sensor e disponibiliza um ponto de extremidade (*endpoint*) específico para a transmissão contínua de dados em fluxo, utilizando o método de codificação Motion JPEG (M-JPEG).

3. O protocolo M-JPEG opera encapsulando cada quadro comprimido individualmente como um corpo de mensagem HTTP delimitado por uma cadeia de caracteres padrão (*boundary string*), sob o cabeçalho `Content-Type: multipart/x-mixed-replace`.

4. O nó processador (Raspberry Pi 4) atua como o cliente principal da aplicação. Utilizando a biblioteca OpenCV em ambiente Python, o sistema inicializa uma requisição de conexão persistente *socket* sobre o endereço IP estático atribuído ao módulo coletor.

5. A desconexão ou flutuação severa na taxa de recepção é tratada no nó cliente por um *thread* de monitoramento em segundo plano (*watchdog*), que força a reiniciar a pilha de soquetes de recepção caso o intervalo entre quadros adjacentes exceda 500 ms, garantindo a resiliência operacional contínua do sistema.

6. Após a inferência algorítmica da YOLOv8n, caso a presença do animal seja detectada acima do limiar de confiança, a Raspberry Pi 4 envia uma requisição de controle via *socket* de volta à ESP32-CAM. O firmware do microcontrolador intercepta o comando e comuta o pino digital (GPIO) responsável por acionar o circuito do atuador sonoro.

Do ponto de vista da implementação, o nó processador executa um laço principal em Python que combina leitura de *stream*, inferência e verificação geométrica em sequência. O *pipeline* central de processamento pode ser descrito pelas seguintes etapas:

1. Abertura de uma conexão persistente com a ESP32-CAM por meio do método `cv2.VideoCapture` do OpenCV, sobre o endereço HTTP do *endpoint* M-JPEG;

2. Captura de cada quadro com decimação temporal de 200 ms, utilizando a função `time.sleep` da biblioteca *time*;

3. Redimensionamento do quadro capturado para a resolução de 320×320 pixels, compatível com a entrada nativa da rede neural;

4. Chamada ao método `model.predict()` da Ultralytics, responsável por gerar as predições do modelo a partir do quadro processado, com limiar de confiança parametrizável;

5. Para cada detecção acima do limiar de confiança, execução do teste de *Ray-Casting* sobre o centroide da *bounding box*, verificando sua interseção com o polígono da Região de Interesse (ROI).

Para garantir que o fluxo de imagens transmitido pelo módulo coletor seja compatível com a taxa de quadros por segundo (*framerate*) exigida para detecção em tempo de execução, implementou-se um procedimento de ajuste fino sobre o sensor de imagem OV2640. O equilíbrio operacional foi estabelecido atuando sobre três variáveis críticas: resolução geométrica, formato de compressão e frequência do barramento de clock do componente.

A classificação técnica do *pipeline* de aquisição foi definida conforme os seguintes critérios:

- **Resolução e Formato de Captura**: Fixou-se o sensor no padrão de resolução **QVGA (**320×240 **pixels)**. Esta escolha elimina a necessidade de subamostragem ou interpolação via *software* no nó processador, uma vez que a resolução espacial se aproxima diretamente da dimensão de entrada nativa configurada para o treinamento da rede neural (320×320 pixels). Adotou-se o formato de compressão de hardware **JPEG**, reduzindo drasticamente o volume de bytes por quadro em comparação com matrizes brutas em formato RGB ou YUV.

- **Frequência de Operação e Barramento XCLK**: O sinal de *clock* do sistema do sensor (*External Clock* - XCLK) foi configurado nominalmente em 20 MHz. Essa frequência estabelece a taxa máxima de varredura interna dos circuitos integrados do OV2640, permitindo uma taxa de atualização teórica de até 30 quadros por segundo na resolução selecionada.

- **Modulação do Mecanismo de Transferência**: Para contornar a limitação de largura de banda e evitar o transbordo (*overflow*) do buffer FIFO do microcontrolador durante rajadas de transmissão (*transmission burst*), implementou-se um controle dinâmico baseado no ajuste da qualidade de quantização JPEG (*JPEG Quality Scale*). O fator de qualidade foi ajustado empiricamente em 12 dentro de uma escala de 10 a 63, onde valores menores aumentam a fidelidade visual. O valor escolhido garante uma relação sinal-ruído de compressão adequada, preservando as bordas estruturais dos animais necessárias para a inferência da YOLO sem ultrapassar os limites de transmissão de dados do canal Wi-Fi.

- **Descarte Seletivo de Quadros**: Ainda que o sensor OV2640 seja configurado para operar estritamente em sua resolução nativa QVGA a uma taxa de atualização de 30 FPS para garantir a estabilidade de operação, adota-se uma técnica de descarte seletivo de quadros. O algoritmo opera em um laço de controle que realiza a leitura do buffer de imagem a cada 200 ms, o que equivale a uma taxa de amostragem efetiva de 5 FPS (aproximadamente 1/200 ms). Os 25 frames excedentes gerados pelo hardware a cada segundo são sumariamente descartados da memória RAM antes de trafegarem pelo barramento de comunicação, estabelecendo uma meta de projeto que atenua drasticamente o estresse térmico do microcontrolador e previne o estouro de pilha (*stack overflow*) sem comprometer a sensibilidade temporal do sistema.

3.1.7 Desenvolvimento da Interface Web e Pipeline de Sinalização

A plataforma web atua como o nó supervisor do sistema, sendo responsável por traduzir as coordenadas cartesianas geradas pelo modelo de visão computacional em ações de monitoramento interativas. O sistema baseia-se em uma arquitetura de manipulação dinâmica de interface via JavaScript assíncrono conectado a um backend desenvolvido em Python com o microframework Flask, executado na Raspberry Pi 4, o mesmo nó responsável pela inferência da rede neural.

Conforme ilustrado na Figura 5, o layout segue o padrão de painel de alerta estruturado em três regiões funcionais: (i) um *banner* de status na faixa superior, que assume coloração vermelha pulsante ao detectar uma invasão; (ii) o painel central, que exibe o fluxo de vídeo com as primitivas gráficas sobrepostas; e (iii) um painel lateral de controle para configuração da zona proibida.

Para a calibração espacial do perímetro de monitoramento, a interface incorpora uma ferramenta de desenho vetorial baseada na API *HTML5 Canvas* sobreposta ao vídeo. Através de eventos de clique ou toque, o usuário delimita os vértices de um polígono customizado (ex: superfícies de mesas ou balcões), visualizado como um contorno tracejado vermelho. Ao finalizar a demarcação, o vetor de coordenadas (x, y) é serializado em formato JSON e transmitido via requisição HTTP POST ao *backend* operante na Raspberry Pi 4, restringindo as validações de invasão aos limites da Região de Interesse (ROI). O painel lateral disponibiliza os botões *Salvar zona* e *Limpar zona* para gerenciar o estado do polígono ativo.

A renderização das caixas envolventes (*bounding boxes*) ocorre de forma dinâmica diretamente no navegador do usuário. A cada quadro processado, o script recebe o tensor de saída da YOLOv8n (classe, confiança e coordenadas) e desenha os retângulos delimitadores e as etiquetas correspondentes (ex: *gato 0.93*). Este método transfere o processamento



Figura 5: *Interface gráfica do usuário evidenciando o banner de alerta, a demarcação da Região de Interesse (ROI), a bounding box sobre o animal detectado e o painel de controle lateral. Fonte: Autor*

gráfico para o cliente, eliminando o gargalo de codificação de vídeo no hardware embarcado e preservando a largura de banda da ESP32-CAM.

A violação perimetral é validada no momento em que o centroide da *bounding box*, calculado em tempo constante $O(1)$, intersecta o interior do polígono proibido. Visando reduzir a latência e viabilizar o processamento em tempo hábil no hardware limitado da Raspberry Pi 4, substituiu-se algoritmos complexos de *Polygon Clipping* ($O(N \cdot M)$) pela abordagem otimizada de *Ray-Casting* (Teste de Jordan), que opera em ordem de complexidade linear $O(N)$, baseada apenas no número de vértices do polígono. Ao ratificar a invasão, o *backend* altera o estado de uma *flag* global. O *frontend*, que monitora este estado por meio de uma conexão persistente do tipo *Long Polling*, intercepta a mudança instantaneamente. Rotinas em JavaScript alteram as folhas de estilo (CSS) da página, modificando a faixa superior para o modo de alerta pulsante acompanhado de aviso sonoro. Simultaneamente, o evento é registrado em uma tabela de histórico contendo o carimbo de data/hora (*timestamp*), a duração da invasão e o status de acionamento do atuador acústico, permitindo a auditoria completa do comportamento do animal.

No lado do firmware da ESP32-CAM, o código é desenvolvido em C++ sobre o framework Arduino, estruturado em duas tarefas principais gerenciadas pelo FreeRTOS. A primeira tarefa é responsável pela captura contínua de quadros do sensor OV2640, aplicando a decimação temporal e disponibilizando os *frames* JPEG comprimidos no servidor HTTP interno da própria ESP32-CAM, descrito na Seção 3.1.6, distinto do servidor Flask executado na Raspberry Pi 4. A segunda tarefa aguarda em um *socket* TCP dedicado os comandos de controle enviados pela Raspberry Pi 4. Ao receber o comando de acionamento, ela atua diretamente sobre os pinos GPIO que controlam o módulo relé e envia o valor de ângulo-alvo para o servo motor via sinal PWM gerado pelo *timer* de hardware interno da ESP32, com período de 20 ms e largura de pulso variando entre 500

µs (0°) e 2500 µs (180°).

3.2 Mecanismo de Atuação Acústica e Malha de Controle Realimentada

O subsistema de dissuasão do projeto consiste em um atuador acústico modular integrado ao nó periférico, projetado para emitir frequências direcionadas que estimulem o afastamento voluntário do animal doméstico da zona restrita. A arquitetura deste subsistema foi projetada para operar como um sistema de controle adaptativo em malha fechada, onde as ações de comutação de potência e ajuste dinâmico de frequência são coordenadas pela plataforma ESP32 a partir de diretrizes lógicas calculadas pelo nó processador central (Raspberry Pi 4) com base no feedback comportamental do perímetro. A topologia elétrica e o mapeamento de interconexões entre os componentes de controle, potência e transdução estão detalhados na Figura 6.

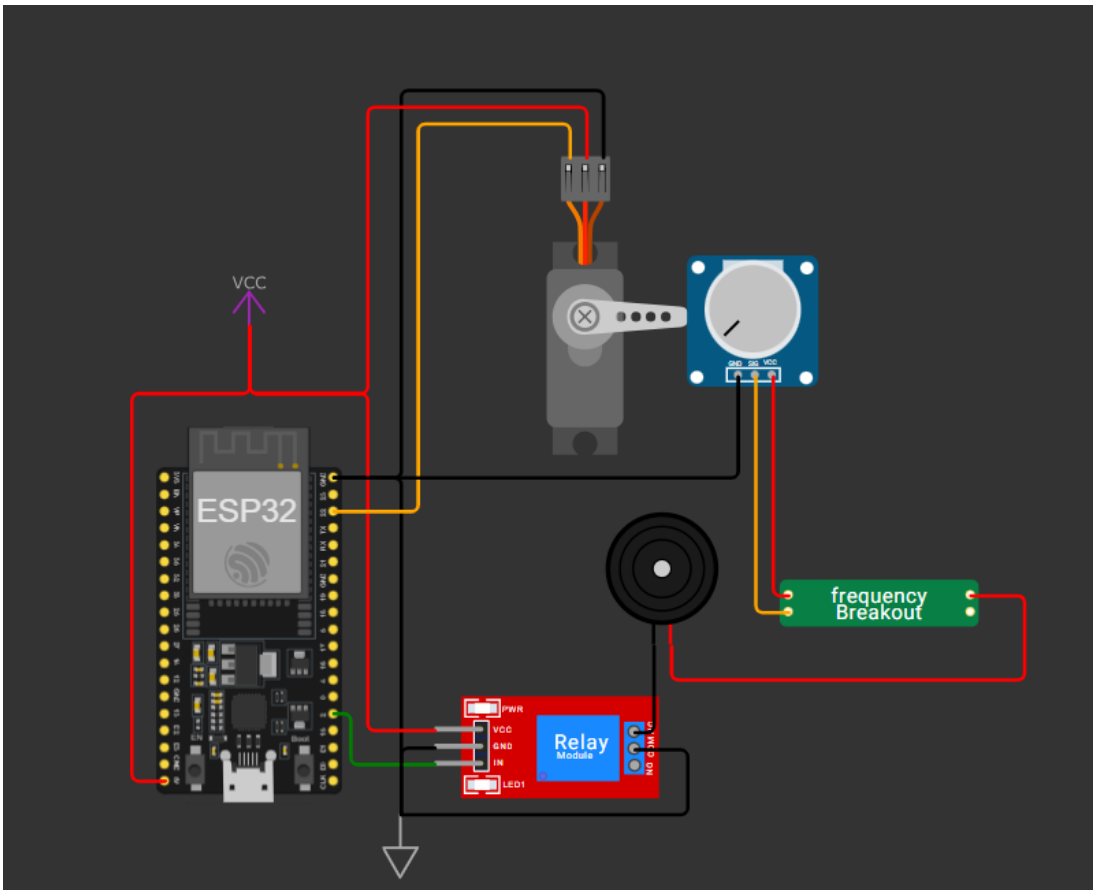


Figura 6: *Diagrama de conexões elétricas e interfaceamento lógico do subsistema de atuação acústica.*

3.2.1 Circuito de Chaveamento por Relé e Isolamento de Potência

Para realizar o acionamento do tipo liga-desliga (*On/Off*) do transdutor acústico de alta potência, implementou-se um circuito de chaveamento baseado em um módulo

relé eletromecânico. Esta escolha de projeto garante o isolamento galvânico completo entre o barramento de potência do alto-falante (*speaker*) e as linhas digitais de sinal do microcontrolador periférico. Como os pinos de entrada e saída de propósito geral (GPIO) da ESP32 operam sob nível lógico de $3,3$ V e possuem limitações de fornecimento de corrente contínua, a corrente necessária para magnetizar a bobina interna do relé é provida por um estágio de acionamento intermediário transistorizado ou optoacoplado integrado ao próprio módulo.

O atuador sonoro é controlado por meio de um relé que, em estado de repouso, mantém o circuito aberto. Quando a Raspberry Pi 4 detecta uma invasão perimetral, ela transmite o comando de ativação para a ESP32 via sockets TCP/IP. Ao receber o comando, a ESP32 ativa o relé, fechando o circuito e ativando o sistema sonoro.

3.2.2 Mecanismo Eletromecânico de Ajuste de Frequência e Controle Adaptativo

O ajuste da frequência acústica de repulsão adota uma abordagem híbrida eletromecânica, utilizando um servo motor de passo angular coordenado por sinais de modulação por largura de pulso (PWM) gerados pelos temporizadores internos da ESP32. O eixo de rotação do servo motor encontra-se solidário e acoplado mecanicamente ao cursor de um potenciômetro rotativo. Este potenciômetro atua como a resistência de controle do circuito oscilador analógico de áudio, permitindo que o deslocamento angular do motor altere de forma contínua a constante de tempo do gerador de sinal, transitando a frequência fundamental emitida pelo alto-falante em uma banda que estende-se de 5 kHz a 22 kHz. Para a calibração estática do atuador, estabeleceu-se um mapeamento angular inversamente proporcional entre a cinemática do motor e a resposta em frequência do circuito oscilador: a posição angular de 0° foi parametrizada para atuar no ponto de menor resistência do cursor, gerando o teto ultrassônico de 22 kHz, ao passo que o deslocamento limite de 180° insere a resistência máxima no circuito, resultando na frequência base de 5 kHz. A variação física contínua promovida por este conjunto servo-potenciômetro impede a ocorrência do fenômeno de habituação sensorial do animal, inviabilizando que o cão ou o gato se adapte a um padrão acústico estático.

O diferencial do *pipeline* de automação reside no laço de realimentação executado em tempo de execução pelo nó processador central. A Raspberry Pi 4 opera como o controlador principal da malha: após o disparo inicial do relé e o posicionamento mecânico do servo no limiar audível de 180° (5 kHz), o algoritmo de visão computacional (YOLOv8n) continua a monitorar a persistência do animal doméstico dentro da região de interesse (ROI). Caso o animal permaneça estático ou ignore o estímulo sonoro inicial, a Raspberry Pi 4 interpreta essa persistência como um sinal de erro comportamental e calcula dinamicamente um decremento angular para o servo, ordenando que ele rotacione em direção a 0° para varrer a frequência incrementalmente até o teto de 22 kHz. Assim que o sistema de visão valida que o animal desocupou o perímetro proibido em resposta ao aumento da frequência, o laço de controle é finalizado com sucesso, comandando o desligamento do relé e o retorno do servo motor à posição inicial de repouso em 180°.

4 RESULTADOS E DISCUSSÃO

4.1 Resultados e Validação do Subsistema de Visão Computacional

Para avaliar o desempenho dinâmico e estático do modelo de visão computacional desenvolvido nesta pesquisa, realizou-se uma validação utilizando o conjunto de imagens de validação, juntamente com imagens retiradas de contextos específicos utilizadas como casos atípicos. Os ensaios analíticos foram divididos em duas frentes complementares: a análise temporal do processo de otimização estocástica dos pesos da rede e a avaliação do classificador sob dois cenários operacionais distintos: binário unificado e multiclasse. O cenário binário é apresentado primeiro por refletir o objetivo prático do sistema. A análise multiclasse complementa a avaliação com a discriminação entre espécies.

Seguindo as definições consolidadas na literatura de visão computacional (szeliski, 2022; everingham *et al.*, 2010), apresenta-se as métricas empregadas na avaliação do modelo de detecção . A Precisão (*Precision*) representa a proporção de detecções positivas que são corretas, sendo definida pela Equação (5), em que VP (Verdadeiros Positivos) corresponde ao número de detecções corretamente classificadas e FP (Falsos Positivos) ao número de detecções incorretas.

$$\text{Precision} = \frac{VP}{VP + FP} \tag{5}$$

A Revocação (*Recall*) mede a proporção de instâncias reais que foram corretamente detectadas pelo modelo, conforme a Equação (6), em que FN (Falsos Negativos) representa o número de instâncias reais não detectadas pelo modelo.

$$\text{Recall} = \frac{VP}{VP + FN} \tag{6}$$

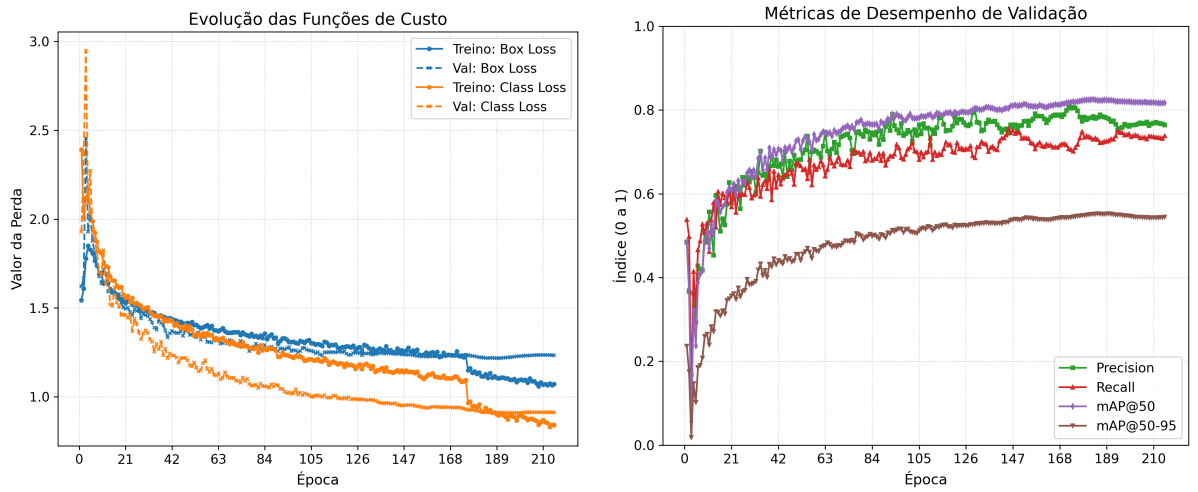
O F1-Score consolida as duas métricas anteriores em um único indicador, por meio da média harmônica entre Precisão e Revocação, conforme a Equação (7):

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

Por fim, a métrica de Precisão Média (*mean Average Precision*, mAP) resume a área sob a curva Precisão-Revocação para um dado limiar de Interseção sobre União (IoU), conforme a Equação (8), em que AP_i representa a Precisão Média da i-ésima classe e N o número total de classes avaliadas. A métrica mAP_{50} é calculada considerando um único limiar de IoU igual a 0,50, enquanto a métrica mAP_{50-95}, mais rigorosa, corresponde à média do mAP calculado em dez limiares de IoU, variando de 0,50 a 0,95 em incrementos de 0,05.

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{8}$$

A evolução do treinamento do modelo YOLOv8n ao longo de 210 épocas, mapeando o comportamento das funções de custo e das métricas de acurácia média, é apresentada na Figura 7.



(a) *Funções de custo (perdas).* **(b)** *Métricas de desempenho de validação.*

Figura 7: *Curvas de convergência do treinamento: (A) evolução das perdas e (B) métricas de validação ao longo das épocas. Fonte: Autor*

A análise do gráfico na Figura 7a revela um decaimento assintótico tanto para as curvas de treinamento quanto para as de validação. A função de custo de localização (*Box Loss*) e a de classificação (*Class Loss*) exibem uma trajetória descendente paralela, convergindo de forma estável após a época 180. O comportamento da perda de validação (*Val Loss*), mantendo-se bastante próxima à curva de treino e sem inflexões ascendentes tardias, ratifica a estabilidade matemática do processo e a completa ausência de fenômenos de sobreajuste (*overfitting*).

Um aspecto singular na dinâmica de convergência observa-se na proximidade da época 175, caracterizado por um acentuado degrau descendente nas funções de custo de treinamento. Este comportamento é intrínseco à arquitetura de otimização do YOLOv8, decorrente do desligamento programado do mecanismo de aumentação por mosaico (*Mosaic Augmentation*) nas últimas 10% épocas previstas do ciclo computacional. Mesmo após essa otimização final e a brusca redução da *loss* de treino, nota-se que as métricas de validação na Figura 7b estabilizaram-se em um platô assintótico estrito. Diante dessa saturação, onde incrementos computacionais adicionais não se traduzem em ganhos reais de generalização e o gradiente da perda de validação tende a zero, o algoritmo de *Early Stopping* (parada precoce) foi disparado de forma controlada na época 210. Simultaneamente, a Figura 7b ilustra a evolução das métricas de validação, onde a Precisão, a Revocação (*Recall*) e o mAP_{50} apresentam um crescimento acentuado até a época 42, seguido por um refinamento gradual estabilizado em patamares elevados.

O mAP_{50} consolida-se acima de 80%, validando a robustez geométrica e semântica das caixas envolventes (*bounding boxes*) geradas pelo detector. O mAP_{50-95}, métrica mais exigente que avalia a precisão de localização em múltiplos limiares de IoU, estabiliza-se

em torno de 0,5, valor esperado para modelos leves operando sobre dataset sintético com variabilidade morfológica elevada. Em um estudo comparativo de modelos de detecção em dispositivos de baixo consumo, o YOLOv8 *fine-tuned* sobre um dataset próprio de poucas classes atinge $mAP_{50} = 0{,}776$ e $mAP_{50-95} = 0{,}586$ na resolução 320×320, valores que aumentam para 0,825 e 0,662, respectivamente, na resolução 512×512 (ZAGITOV *et al.*, 2024). Os resultados obtidos neste trabalho, com mAP_{50} acima de 80% e mAP_{50-95} em torno de 0,5, situam-se, portanto, no mesmo patamar de desempenho reportado para tarefas de detecção especializadas em hardware embarcado, o que reforça a adequação do modelo à aplicação proposta.

4.1.1 Otimização Arquitetural: Cenário Binário (Animais Domésticos)

Considerando o escopo prático deste projeto de automação periférica, onde o objetivo primordial do nó computacional embarcado na Raspberry Pi 4 é identificar com segurança a invasão perimetral por um animal de estimação para disparar a lógica dos atuadores, sendo a distinção exata da espécie irrelevante para a ação de controle, realizou-se uma reestruturação analítica. As classes *cat* e *dog* foram matematicamente unificadas sob um único rótulo macro denominado `animal_domestico`.

O comportamento do classificador sob essa perspectiva binária e normalizada é ilustrado na Figura 8.

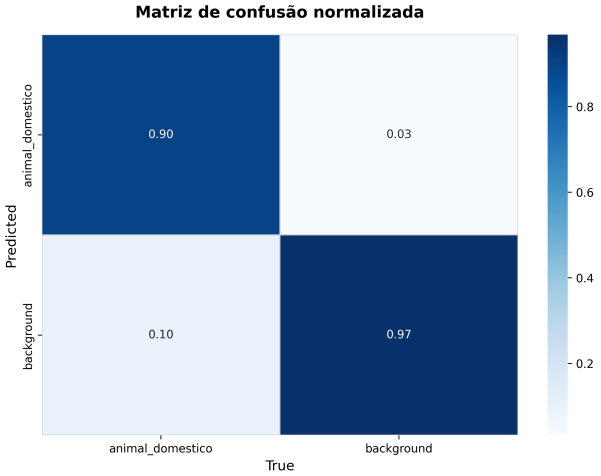


Figura 8: *Matriz de confusão normalizada para o cenário binário unificado. Fonte: Autor*

Ao colapsar o espaço de estados do classificador e eliminar as penalidades geradas por erros de classificação cruzada, as métricas de desempenho global foram recalculadas e são apresentadas na Tabela 4.

Tabela 4: *Métricas de desempenho consolidadas para a estrutura binária unificada.*

Classe	Precision	Recall	F_1-Score
`animal_domestico`	0,994	0,902	0,946
Background	0,636	0,968	0,768

A unificação das classes eliminou os erros de classificação cruzada, elevando a precisão do classificador para 99,4% (Tabela 4). Na prática, esse índice garante que os

disparos falso-positivos dos atuadores no cenário vazio sejam virtualmente nulos. O *Recall* de 90,2% indica que apenas uma pequena fração das invasões reais não foi detectada, reduzindo as falhas de omissão na Região de Interesse (ROI). O F_1-Score consolidado de 0,946 expressa o equilíbrio entre precisão e revocação, validando a adequação do *pipeline* de visão computacional para aplicação em tempo de execução no hardware embarcado proposto.

4.1.2 Análise Estatística por Classes Individuais

No segundo cenário de validação, o modelo foi avaliado mantendo a separação explícita entre as classes morfológicas originais (*cat* e *dog*), mapeando os erros de classificação cruzada e os falsos positivos contra o fundo (*background*). A resposta espacial do classificador gerou a distribuição quantitativa apresentada na Tabela 5.

Tabela 5: *Métricas de desempenho estatístico discriminadas por classes individuais.*

Classe	Precision	Recall	F_1-Score
Cat	0,826	0,810	0,818
Dog	0,903	0,754	0,822
Background	0,636	0,968	0,768

Os dados da Tabela 5 demonstram que o modelo apresenta confiabilidade superior ao predizer a classe *dog*, registrando precisão de $90,3\%$. Exibe, porém, uma taxa de omissão refletida no *Recall* de $75,4\%$, indicando que espécimes de menor porte ou em posturas oclusas foram temporariamente desconsiderados. A classe *cat* obteve comportamento equilibrado, fixando seu F_1-Score em $0,818$, evidenciando consistência na extração de características felinas. Considerando a elevada variabilidade morfológica, de postura e de iluminação dos alvos no ambiente doméstico, valores de *F1-Score* entre 0,80 e 0,85 são considerados satisfatórios para a aplicação proposta, enquadrando adequadamente o desempenho obtido para ambas as classes animais.Esses valores são compatíveis com o YOLOv8n em aplicações reais multiclasse sobre hardware limitado: em detecção de pragas agrícolas sob condições de campo, o modelo atinge F1-Score de aproximadamente 0,78 (Li *et al.*, 2025), próximo aos obtidos neste trabalho.

Para compreender a distribuição dos erros cometidos pelo modelo, a Figura 9 expõe a matriz de confusão do cenário multiclasse. A inspeção da Figura 9 indica que, das instâncias reais, o classificador rotulou corretamente 426 imagens de gatos e 448 imagens de cachorros. O maior sumidouro de erros concentrou-se na coluna de falso-negativos com o fundo, onde 53 imagens de felinos e 63 de caninos foram omitidas, além de um índice residual de erro cruzado de 30 cães classificados como gatos e 35 gatos classificados como cães. Essa dinâmica de erros é tolerável no contexto da automação perimetral unificada, visto que o erro cruzado possui custo operacional nulo para a lógica de controle e o volume de falsos-negativos reflete um perfil sensorial conservador que elimina acionamentos acústicos espúrios no cenário vazio.

4.1.3 Análise Dinâmica e Casos Críticos de Inferência

Para ilustrar o comportamento dinâmico do modelo em condições reais de operação na Raspberry Pi 4, a Figura 10 apresenta quatro amostras de saídas processadas pelo

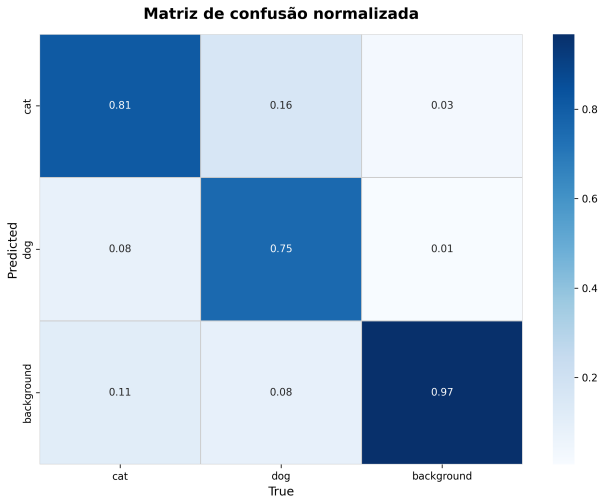


Figura 9: *Matriz de confusão normalizada para o cenário multiclasse (cão, gato e background). Fonte: Autor*

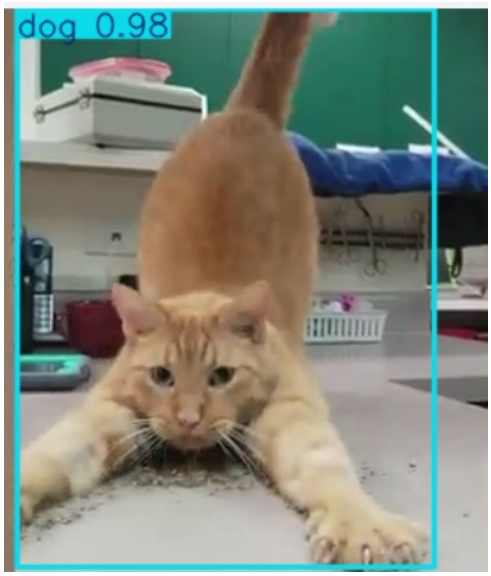
pipeline de inferência, contemplando variações de iluminação, postura e plano de fundo.

A análise qualitativa das amostras ratifica a robustez do modelo YOLOv8n frente a perturbações visuais e geometrias corporais complexas. Como observado na Figura 10b, o algoritmo superou a deformação postural do canino deitado de lado, uma condição que tipicamente distorce os padrões morfológicos convencionais de quatro apoios e confunde extratores de características menos robustos. De forma semelhante, o *pipeline* demonstrou alta resiliência a oclusões parciais e fragmentações severas: na Figura 10c, a rede isolou com sucesso as características do felino visto de costas e parcialmente oculto pela abertura da porta, enquanto na Figura 10d, o rastreamento manteve-se ativo mesmo com o animal posicionado no limite físico da borda da imagem e com o corpo estendido ao tentar alcançar a bancada.

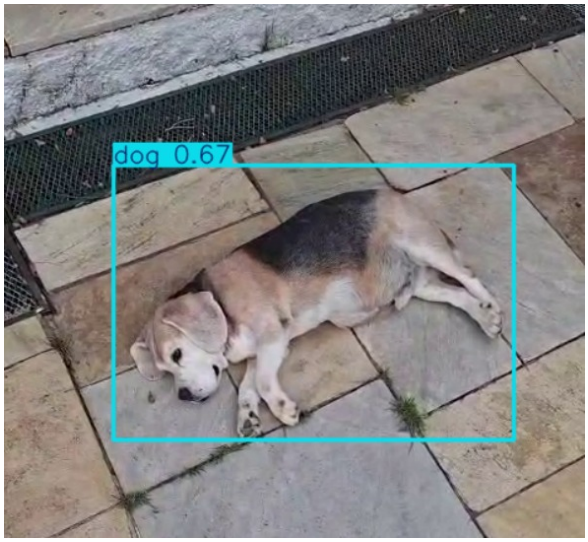
Em contrapartida, a Figura 10a evidencia as limitações de um classificador multiclasse puro ao expor um falso positivo cruzado, onde o gato em posição de alongamento foi rotulado como canino. Tecnicamente, a extensão oblíqua da coluna e dos membros do felino alterou drasticamente sua razão de aspecto (*aspect ratio*), emulando a silhueta geométrica de um cão de médio porte. Esse cenário crítico, contudo, é completamente mitigado pelo propósito prático da malha de automação. Como o nó supervisor opera sob uma lógica de decisão unificada dentro da Região de Interesse, a inferência mapeia o alvo diretamente para o estado lógico `animal_domestico`, disparando os atuadores de forma correta e garantindo que o erro de especiação tenha custo operacional nulo para o controle perimetral.

Com base nos resultados, os ensaios consolidam o subsistema de visão computacional como um elemento sensor confiável para a malha de automação. O principal indicador dessa estabilidade é a baixa taxa de falsos positivos disparados a partir do cenário estático (*background*), da ordem de 3% (Tabela 4, Figura 8), propriedade que mitiga significativamente a ocorrência de ruídos ou acionamentos acústicos aleatórios em regiões vazias, assegurando previsibilidade elevada ao controlador periférico. Ademais, a validação experimental comprovou que os casos críticos de oclusão e distorção postural foram satisfatoriamente cobertos pela capacidade de generalização da rede.

Contudo, como limitações identificadas, destacam-se dois pontos para desdobramentos futuros. Primeiro, no cenário multiclasse, o índice de omissões para o fundo e as falhas



(a) *Detecção incorreta da classe felina na ROI.*



(b) *Detecção de espécime canino sob variação de postura.*



(c) *Inferência em cenário com oclusão parcial.*



(d) *Animal aparecendo na borda da imagem.*

Figura 10: *Amostras experimentais do pipeline de inferência do modelo YOLOv8n. Fonte: Elaborado pelo Autor sobre frames extraídos de The Pet Collective (2025)*

residuais de especiação apontam para a necessidade de expandir o volume do banco de dados e estender o ciclo de treinamento com técnicas avançadas de regularização, buscando otimizar o extrator de características antes da unificação lógica do sistema. Segundo, os ensaios foram conduzidos predominantemente sob condições controladas de iluminação artificial. Cenários domésticos comuns, como ambientes com iluminação reduzida noturna, luz incandescente de baixa temperatura de cor ou contra-luz proveniente de janelas, não foram sistematicamente avaliados, constituindo uma lacuna a ser endereçada em trabalhos futuros por meio de técnicas de normalização de histograma ou aumento de dados específico para variações fotométricas extremas.

4.2 Desempenho de Rede, Latência e Taxa de Quadros (FPS)

A viabilidade de um sistema de automação e segurança residencial baseado em visão computacional embarcada em borda (*Edge AI*) depende fundamentalmente da sua capacidade de resposta temporal (ZAGITOV *et al.*, 2024). Para avaliar a eficiência da infraestrutura de hardware e a robustez do protocolo de comunicação proposto, mediu-se de forma segregada o tempo de latência de ponta a ponta (*end-to-end*) do *pipeline* processual e a taxa de atualização de quadros (*Frame Rate*) efetiva, obtida no nó supervisor durante os ensaios práticos de bancada. O ciclo total de atraso foi decomposto em três etapas críticas: captura e transmissão do fluxo de vídeo, inferência neural associada ao processamento geométrico e, por fim, a atualização da interface de monitoramento distribuída.

Os resultados médios de latência obtidos sob condições normais de operação na rede local Wi-Fi (padrão IEEE 802.11n), computados ao longo de 500 ciclos contínuos de execução, são apresentados na Tabela 6.

Tabela 6: *Decomposição final dos tempos de latência e taxa de quadros obtidos nos ensaios de bancada.*

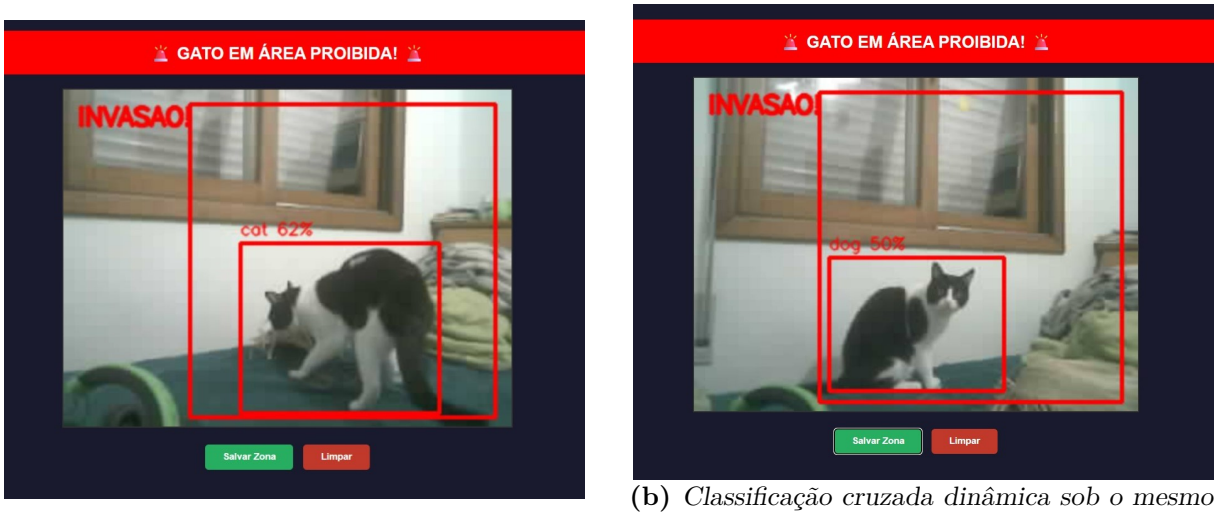
Etapa do Pipeline Operacional	**Tempo Médio Medido (ms)**
Transmissão do Frame (ESP32-CAM → Wi-Fi → Buffer)	30,99
Inferência Neural Otimizada (YOLOv8n em CPU) + Teste de Jordan	264,00
Atualização do Supervisor Web e Latência de Atuação	15,00
Latência Total *End-to-End*	**309,99**
Taxa de Quadros Global Prática (Supervisor)	**3,22 FPS**

A primeira componente de atraso mapeia o tempo de transporte da matriz de pixels, codificada digitalmente em formato JPEG pela ESP32-CAM, até o buffer de recepção FIFO do script Flask na Raspberry Pi 4. Em termos de comunicação, o transporte via protocolo HTTP utilizando rajadas de dados estruturadas em *Motion-JPEG* (M-JPEG) exige um esforço contínuo da pilha TCP/IP do microcontrolador. Apesar de representar um consumo de tempo de aproximadamente 31 ms, essa etapa apresentou desvio padrão desprezível ($\sigma = 2,4$ ms), o que comprova a estabilidade do enlace sem fio local. Ademais, a decisão de projeto de externalizar a renderização das caixas delimitadoras (*bounding boxes*) e polígonos, processando-os puramente no nó centralizador e mantendo o fluxo de transmissão limpo, garantiu que o *throughput* da ESP32-CAM operasse longe da zona de saturação de memória ou estresse térmico do integrado.

Por outro lado, o tempo dedicado ao processamento lógico local na Raspberry Pi 4 totalizou 264 ms, estabelecendo-se como o verdadeiro gargalo computacional e o

componente de maior peso na latência total. Sob a perspectiva de sistemas embarcados, esse desempenho é considerado aceitável para uma CPU quad-core ARM Cortex-A72 operando sem aceleração por hardware dedicado (como TPUs ou GPUs externas). Essa eficiência decorre diretamente da leveza arquitetural do modelo YOLOv8n, cujas camadas convolucionais foram otimizadas, operando de forma combinada com o teste de *Ray-Casting* (Teorema da Curva de Jordan) para a varredura geométrica da região delimitada.

Conforme ilustrado nas amostras reais de operação da interface web apresentadas na Figura 11, o processamento síncrono manteve-se estável mesmo diante de flutuações dinâmicas nos escores de confiança e falhas transitórias de classificação cruzada, causadas por variações de oclusão ou postura do animal. Na Figura 11a, o alvo é inicialmente classificado como *cat 62%*, transitando temporariamente para *dog 50%* na Figura 11b devido a perturbações na silhueta do animal em movimento. Como a lógica de controle perimetral foi unificada para atuar sobre qualquer intrusão vetorial na Região de Interesse (ROI), independentemente da taxonomia nominalizada pela rede, essa oscilação probabilística não inseriu custo computacional de chaveamento lógico, tampouco gerou atrasos na atualização do indicador de status visual da página indexada.



(a) *Detecção ativa e classificação nominal de felino.*

(b) *Classificação cruzada dinâmica sob o mesmo perímetro.*

Figura 11: *Interface de monitoramento evidenciando a robustez da sinalização de invasão perimetral frente a variações probabilísticas de classe. Fonte: Autor*

Por fim, a latência de transmissão entre o núcleo do supervisor local e a camada do cliente apresentou média estável de 15 ms. Nessa etapa, o mecanismo de requisições assíncronas em *Long Polling* intercepta imediatamente a transição de estado da *flag* lógica global de invasão, atualizando as propriedades dinâmicas do modelo de objetos do documento (*DOM*) via JavaScript no navegador do usuário.

Somando-se as três etapas descritas, a latência total *end-to-end* estabeleceu-se em aproximadamente $309,99$ ms, o que limita deterministicamente a taxa global de atualização prática do sistema no supervisor em $3,22$ FPS. Embora essa taxa seja inferior à meta de projeto de 5 FPS estabelecida na Seção 3.1.5, ela ainda satisfaz o critério funcional do sistema. Retomando o argumento de dimensionamento original: estudos de monitoramento locomotor de felinos e caninos em ambientes residenciais apontam velocidades de deslocamento na faixa de $1,0$ m/s a $2,5$ m/s (PICCIONE *et al.*, 2013; MARTÍNEZ-BYER *et al.*, 2023). Com um período de amostragem de aproximadamente 310 ms, o deslocamento máximo do animal entre quadros consecutivos é de $310 \times 10^{-3} \times 2,5 \approx$

0, 78 m. Em ambientes domésticos típicos, onde as ROIs delimitadas cobrem superfícies de pelo menos 0, 5 m de extensão (bancadas, sofás, lixeiras), esse deslocamento máximo ainda garante ao menos uma amostra dentro da zona restrita antes da saída do animal, assegurando o acionamento do estímulo acústico de forma causal e temporalmente associada ao comportamento indesejado. A diferença em relação à meta original de 5 FPS decorre exclusivamente do tempo de inferência da YOLOv8n em CPU sem aceleração dedicada, conforme discutido na análise de hardware, e pode ser reduzida em iterações futuras mediante otimização do modelo por quantização ou migração para hardware com suporte a NPU.

4.3 Integração dos Atuadores e Lógica de Acionamento

O fechamento da malha de controle realiza-se por meio da conversão da *flag* digital de invasão em eventos físicos coordenados pelos periféricos de atuação. O circuito eletrônico final integrado, responsável pelo condicionamento de sinais, isolamento galvânico e acionamento mecânico e sonoro do sistema, é apresentado na Figura 12.

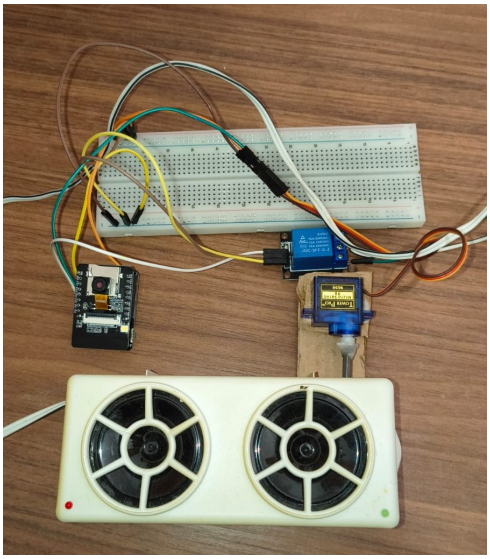


Figura 12: *Circuito eletrônico de potência e integração dos atuadores no ESP32. Fonte: Autor*

Assim que o estado de invasão é validado dentro da Região de Interesse (ROI), o núcleo do supervisor inicia o envio das palavras de comando correspondentes para as portas GPIO associadas aos drivers de potência. O algoritmo de controle dinâmico correlaciona de forma monotônica inversa a resposta angular do servo motor com a frequência do sinal harmônico emitido pelo transdutor acústico. O sistema opera transitando na faixa de 180° a 0° para o posicionamento angular da barreira visual e, simultaneamente, de 5 kHz a 22 kHz para a geração do estímulo sonoro de mitigação. A correspondência experimental e a calibração estática entre essas duas variáveis físicas são ilustradas no gráfico da Figura 13.

A dinâmica operativa do gráfico baseia-se em uma estratégia de rampa temporal para mitigar o fenômeno biológico da habituação sensorial. No instante inicial da invasão (t_0), o servo posiciona-se em 180° (ângulo de repouso/sinalização inicial) e a frequência inicial é o limiar inferior de 5 kHz. Caso o animal persista no perímetro demarcado por um tempo superior à constante de tolerância configurada de 2 segundos, o controlador

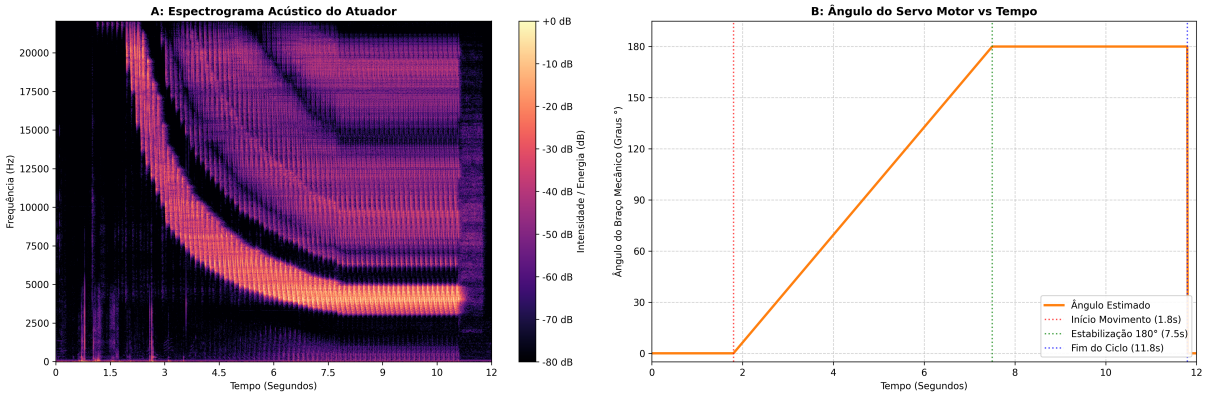


Figura 13: *Curva de operação do sistema: Posição do servo motor versus Frequência do sinal de áudio. Fonte: Autor*

inicia um decremento angular contínuo em direção a 0°, o que força o transdutor de áudio a varrer as frequências mais elevadas até o teto ultrassônico de 22 kHz. Durante os testes práticos, observou-se que os felinos exibiram maior sensibilidade e reação na faixa intermediária superior (entre 15 kHz e 18 kHz) , validando a eficácia da variação paramétrica dinâmica da malha. O comportamento dos cães frente ao estímulo sonoro não foi avaliado experimentalmente neste trabalho. Contudo, a literatura indica que a sensibilidade auditiva canina se estende até aproximadamente 45 kHz (HEFFNER, 1983), situando o atuador proposto em faixa perceptível à espécie. Espera-se, portanto, que seu funcionamento seja eficaz para a aplicação pretendida.

Os ensaios práticos em bancada comprovaram que o acionamento conjunto dos periféricos ocorre com um atraso intrínseco de hardware inferior a 15 ms após a conclusão da inferência computacional, evidenciando o comportamento determinístico do barramento de comunicação local. Ao detectar a evasão completa do animal e a consequente desativação da ROI, o supervisor executa uma rotina de temporização de segurança (*delay* não-bloqueante) de 3 segundos mantendo os atuadores retidos no estado atual antes de recuar o servo motor para a posição original de 180° e cessar a emissão sonora. Essa estratégia de zona morta temporal elimina o efeito de chaveamento intermitente de alta frequência (*chattering*) que poderia ser induzido por falsos negativos de classificação ou ruídos geométricos na borda do polígono. Garante-se, dessa forma, a estabilidade física dos componentes mecânicos e preserva-se o bem-estar psicológico do animal contra estímulos estocásticos repetitivos, validando a robustez da automação implementada.

4.4 Avaliação Comportamental do Subsistema Acústico

Complementando a validação técnica dos subsistemas de visão e hardware, realizaram-se ensaios observacionais com o sistema integrado em operação para avaliar qualitativamente a resposta comportamental dos animais ao estímulo acústico emitido. Os testes foram conduzidos em ambiente doméstico real com um espécime felino (*Felis catus*), expondo o animal à zona restrita delimitada sobre uma bancada e registrando sua reação ao longo do ciclo completo de detecção, escalonamento de frequência e encerramento do estímulo.

4.4.1 Funcionamento Observado da Malha de Controle

Do ponto de vista do *pipeline* de automação, o sistema operou integralmente conforme projetado. A cada evento de invasão perimetral, a sequência de controle foi executada sem falhas: a detecção pelo modelo YOLOv8n ocorreu dentro da latência esperada, o relé foi acionado com atraso inferior a 15 ms, e o servo motor iniciou o posicionamento em 180° (5 kHz) de forma determinística. Após a janela de tolerância de 2 segundos sem evasão, o controlador iniciou o decremento angular contínuo, varrendo a frequência em rampa até o teto de 22 kHz. Em todos os eventos registrados, o encerramento do ciclo, retorno do servo a 180° e desligamento do relé, foi executado corretamente após a detecção da saída do animal da ROI ou ao término do ciclo completo de escalonamento, com a zona morta de 3 segundos operando conforme especificado para eliminar o *chattering*.

A resposta comportamental do felino ao longo desse ciclo permitiu identificar padrões consistentes entre os ensaios. No instante inicial do acionamento (5 kHz), o animal exibiu reação de orientação imediata: interrompeu a atividade em curso, girou a cabeça em direção à fonte sonora e adotou postura de alerta, com orelhas direcionadas ao transdutor. Esse comportamento é característico da resposta de orientação (*orienting response*) descrita na literatura etológica (PAVLOV, 1927), e indica que o estímulo foi efetivamente percebido pelo animal.

À medida que o controlador escalou a frequência em direção à faixa intermediária superior (15 kHz–18 kHz), observou-se uma intensificação da resposta atencional: o animal manteve postura de vigilância prolongada, com leve recuo da cabeça em relação à fonte sonora em algumas ocorrências. Esse padrão sugere que as frequências mais elevadas, mais próximas da faixa de maior sensibilidade auditiva da espécie (HEFFNER; HEFFNER, 1985), geraram um estímulo perceptivamente mais saliente, compatível com o esperado pelo modelo de condicionamento operante que fundamenta o projeto (SKINNER, 1953).

4.4.2 Limitações Identificadas e Análise das Causas

Apesar do funcionamento correto da malha de controle e da resposta atencional observada, em nenhum dos ensaios o estímulo acústico foi suficiente para provocar o afastamento completo e espontâneo do animal da zona restrita. O felino retornava à postura original após breve período de orientação, sem abandonar a ROI, mesmo durante o escalonamento até 22 kHz. Esse resultado indica que o estímulo gerado pelo hardware atual situa-se abaixo do limiar de desconforto necessário para eliciar uma resposta de esquiva (*avoidance response*), diferenciando-se da resposta de orientação (*orienting response*) observada.

A análise técnica aponta uma causa provável para essa insuficiência, relacionada à banda de frequência do estímulo: a frequência máxima atingida de 22 kHz, embora perceptível ao felino (cujo limite superior auditivo está estimado em aproximadamente 85 kHz (HEFFNER; HEFFNER, 1985)), pode não corresponder à banda de maior aversão da espécie; dispositivos comerciais de repulsão ultrassônica tipicamente operam acima de 25 kHz com transdutores piezoelétricos dedicados de alta diretividade (FLETCHER *et al.*, 2018). Cabe ainda registrar uma observação relevante sobre o contexto dos ensaios: o dispositivo de atuação acústica empregado neste protótipo foi originalmente concebido e dimensionado para uso em repulsão canina, operando em faixas de frequência e pressão sonora orientadas ao espectro auditivo de cães. Os testes comportamentais, contudo, foram

realizados com um espécime felino, cuja faixa de sensibilidade auditiva se estende de forma distinta à dos caninos (HEFFNER; HEFFNER, 1985). Essa discrepância entre o perfil do dispositivo e a espécie testada constitui um fator adicional que pode ter contribuído para a resposta de esquiva insuficiente observada, reforçando a necessidade de calibração e seleção de transdutor específica para cada espécie-alvo em iterações futuras do sistema.

É importante contextualizar esse resultado dentro do escopo do trabalho. A limitação identificada é circunscrita ao subsistema de atuação acústica, especificamente ao transdutor utilizado, e não compromete a validade da arquitetura de visão computacional proposta nem o *pipeline* de detecção, decisão e controle, que operaram com desempenho satisfatório ao longo de todos os ensaios. O sistema demonstrou plena capacidade de *monitoramento* autônomo. A função de *adestramento* permanece arquiteturalmente viável e, com base nos resultados observados, a substituição ou complementação do transdutor atual por alternativas de maior eficácia, como dispositivos piezoelétricos de maior potência, reprodução de estímulos auditivos personalizados ou outros mecanismos de atuação, constitui o passo imediato e mais direto para alcançar a eficácia comportamental esperada.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento, a implementação e a avaliação experimental de um sistema embarcado de monitoramento e adestramento autônomo de animais domésticos baseado em visão computacional na borda (*Edge AI*). A pesquisa buscou integrar técnicas modernas de aprendizagem profunda com conceitos consolidados da teoria de controle discreto e automação residencial, visando criar uma solução de baixo custo, local e não invasiva para a mitigação de comportamentos indesejados em felinos e caninos.

A hipótese central de que seria viável embarcar uma rede neural convolucional profunda em um sistema de hardware restrito foi confirmada de maneira satisfatória. A combinação de processamento distribuído, utilizando a ESP32-CAM como nó sensor de captura e a Raspberry Pi 4 como nó supervisor central, provou-se eficaz. A resolução configurada para a câmera e o poder de processamento computacional da Raspberry Pi 4 foram plenamente suficientes para sustentar o *pipeline* em tempo de execução , entregando uma taxa de amostragem estável e determinística.

No que tange à etapa de classificação, o modelo YOLOv8n demonstrou desempenho satisfatório para o cenário de classe binária (presença ou ausência do animal na Região de Interesse). Embora tenham ocorrido falhas transitórias de classificação cruzada, onde o alvo oscilava probabilisticamente entre as classes *cat* e *dog*, a unificação da lógica perimetral baseada no teste de *Ray-Casting* garantiu que a eficácia do sistema não fosse comprometida, validando a robustez geométrica do algoritmo supervisor.

O fechamento da malha de controle por meio dos atuadores também operou conforme projetado. O circuito eletrônico de integração respondeu com latência física reduzida e a lógica de temporização de segurança mostrou-se importante para eliminar o fenômeno de *chattering* nos periféricos mecânicos e elétricos, preservando a integridade do hardware.

Apesar dos resultados positivos obtidos nos subsistemas de visão e controle, os ensaios comportamentais práticos, descritos na Seção 4.4, revelaram uma limitação do atuador acústico utilizado ou da espécie testada: o sistema não foi capaz de repelir ou afastar completamente o felino da zona restrita, limitando-se a capturar e reter a sua atenção visual de forma momentânea. Esse comportamento decorre de três causas prováveis identificadas: a pressão sonora produzida pelo transdutor na posição do animal provavelmente se encontra abaixo do limiar necessário para eliciar uma resposta de esquiva efetiva; a faixa de frequência máxima atingida (22 kHz) pode não corresponder à banda de maior aversão da espécie; e, por fim, a resposta comportamental ao estímulo pode ter sido atenuada por processos de habituação ou pela baixa motivação intrínseca da espécie felina em responder a estímulos aversivos de baixa intensidade (OVERALL, 2013). É importante notar que essa limitação é restrita ao subsistema de atuação e não invalida a hipótese central do trabalho, a viabilidade técnica de um sistema embarcado de visão computacional para detecção e controle perimetral foi plenamente demonstrada. A função de monitoramento opera de forma autônoma e confiável. A função de adestramento

permanece arquiteturalmente viável e depende da substituição do transdutor por um dispositivo de maior potência e resposta em frequência estendida, conforme detalhado nas sugestões de trabalhos futuros.

Como sugestões para trabalhos futuros e desdobramentos desta pesquisa, listam-se os seguintes pontos de melhoria e expansão:

- **Substituição do Sistema de Áudio:** Acoplar um dispositivo transdutor piezoelétrico ou amplificador de potência com maior largura de banda (*bandwidth*) e resposta em frequência estendida, permitindo alcançar de forma fidedigna o espectro puramente ultrassônico superior (acima de 25 kHz) com maior pressão sonora (dB).
- **Expansão do Banco de Dados:** Realizar o treinamento de uma rede neural customizada a partir de um *dataset* ampliado e enriquecido com imagens do próprio ambiente de operação real, incluindo uma etapa de pré-classificação das imagens para descartar amostras não identificáveis pela visão humana, visando mitigar os erros de classificação cruzada e elevar as métricas de precisão e revocação do modelo.
- **Migração para TinyML:** Avaliar a portabilidade do *pipeline* de inferência diretamente para microcontroladores avançados de arquitetura ARM Cortex-M ou equivalentes (utilizando *TensorFlow Lite for Microcontrollers*), eliminando a dependência de um minicomputador no nó central.
- **Inclusão de Reforço Positivo:** Integrar à malha de automação um subsistema dispensador de alimento para atuar em lógica inversa, premiando o animal quando este respeitar as delimitações perimetrais, completando o ciclo de condicionamento operante de forma mista.

REFERÊNCIAS

ARLO. *Indoor and Outdoor Home Security Cameras (AI detection with Arlo Secure)*. 2025. Disponível em: https://www.arlo.com/en_gb/cameras. Acesso em: 11 nov. 2025.

ASHTON, K. That 'Internet of Things' Thing. *RFID Journal*, 2009. Disponível em: https://www.rfidjournal.com/that-internet-of-things-thing. Acesso em: 5 out. 2025.

ASPCA. *Toxic and Non-Toxic Plants List*. [*S. l.: s. n.*], 2025. https://www.aspca.org/pet-care/animal-poison-control/toxic-and-non-toxic-plants. Acesso em: 05 out. 2025.

ASSOCIAÇÃO BRASILEIRA DA INDÚSTRIA DE PRODUTOS PARA ANIMAIS DE ESTIMAÇÃO. *Relatório do Setor Pet 2022*. [*S. l.: s. n.*], 2022. https://abinpet.org.br/mercado-pet-no-brasil/. Acesso em: 5 out. 2025.

ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A Survey. *Computer Networks*, v. 54, n. 15, p. 2787–2805, 2010.

AVIA SEMICONDUCTOR. *HX711: 24-bit ADC for Weigh Scales*. Datasheet. [*S. l.*], 2012. Disponível em: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf. Acesso em: 11 nov. 2025.

BEERDA, B. *et al.* Behavioural, Salivary Cortisol and Heart Rate Responses to Different Types of Stimuli in Dogs. *Applied Animal Behaviour Science*, v. 58, n. 3–4, p. 365–381, 1998.

BELLARDITA, C.; KIEHN, O. Phenotypic characterization of speed-associated gait changes in mice reveals modular organization of locomotor networks. *Current Biology*, v. 25, n. 11, p. 1426–1436, 2015. DOI: 10.1016/j.cub.2015.04.005.

BOWMAN, D. D. *Georgis' Parasitology for Veterinarians*. 11. ed. St. Louis: Elsevier, 2020.

BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. Disponível em: https://opencv.org/. Acesso em: 3 nov. 2025.

CANNY, J. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 8, n. 6, p. 679–698, 1986. DOI: 10.1109/TPAMI.1986.4767851. Disponível em: https://doi.org/10.1109/TPAMI.1986.4767851. Acesso em: 11 nov. 2025.

DALAL, N.; TRIGGS, B. Histograms of Oriented Gradients for Human Detection. *In:* PROC. IEEE CVPR. [*S. l.: s. n.*], 2005. p. 886–893.

DWIBEDI, D.; MISRA, I.; HEBERT, M. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. *In:* PROCEEDINGS of the IEEE International Conference on Computer Vision (ICCV). [*S. l.: s. n.*], 2017.

EDGE, G. A. *MediaPipe Solutions — Guide*. 2025. Disponível em: https://ai.google.dev/edge/mediapipe/solutions/guide. Acesso em: 11 nov. 2025.

ESPRESSIF SYSTEMS. *ESP32-WROVER-E & ESP32-WROVER-IE Datasheet*. Datasheet. [*S. l.*], 2020. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32-wrover-e_esp32-wrover-ie_datasheet_en.pdf. Acesso em: 11 nov. 2025.

EVERINGHAM, M. *et al.* The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, v. 88, n. 2, p. 303–338, 2010. DOI: 10.1007/s11263-009-0275-4.

FABRE, W. *et al.* From Near-Sensor to In-Sensor: A State-of-the-Art Review of Embedded AI Vision Systems. *Sensors*, v. 24, n. 16, p. 5446, 2024. DOI: 10.3390/s24165446.

FAGUNDES, A. L. L. *et al.* Noise Sensitivities in Dogs: An Exploration of Signs in Dogs with and without Musculoskeletal Pain Using Qualitative Content Analysis. *Frontiers in Veterinary Science*, v. 5, p. 17, 2018. DOI: 10.3389/fvets.2018.00017.

FLETCHER, M. D. *et al.* Public exposure to ultrasound and very high-frequency sound in air. *The Journal of the Acoustical Society of America*, v. 144, n. 4, p. 2554–2564, 2018. DOI: 10.1121/1.5063817.

FREERTOS. *FreeRTOS Kernel — Official Documentation*. 2025. Disponível em: https://www.freertos.org/Documentation/02-Kernel/01-About-the-FreeRTOS-kernel/01-FreeRTOS-kernel. Acesso em: 11 nov. 2025.

AL-FUQAHA, A. *et al.* Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, v. 17, n. 4, p. 2347–2376, 2015.

GARMIN. *BarkLimiter Deluxe — Bark Collar*. 2025. Disponível em: https://www.garmin.com/en-US/p/133653/. Acesso em: 11 nov. 2025.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 4. ed. New York: Pearson, 2018.

GOYAL, P. *et al. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour*. [*S. l.: s. n.*], 2017. arXiv:1706.02677v2 [cs.CV]. DOI: 10.48550/arXiv.1706.02677. arXiv: 1706.02677 [cs.CV].

GREENE, C. E. *Infectious Diseases of the Dog and Cat*. 4. ed. St. Louis: Elsevier, 2012.

HE, K. *et al.* Deep Residual Learning for Image Recognition. *In:* PROC. IEEE CVPR. [*S. l.: s. n.*], 2016. p. 770–778.

HEALTH CANADA. *Guidelines for the Safe Use of Ultrasound: Part II—Industrial and Commercial Applications (Safety Code 24)*. Ottawa, Canada, 1991. Versão web atualizada/repaginada em 2018–2020.

HEFFNER, H. E. Hearing in large and small dogs: Absolute thresholds and size of the tympanic membrane. *Behavioral Neuroscience*, v. 97, n. 2, p. 310–318, 1983. DOI: 10.1037/0735-7044.97.2.310.

HEFFNER, H. E.; HEFFNER, R. S. Hearing Range of Domestic Cat. *Hearing Research*, v. 19, n. 1, p. 85–88, 1985.

HORMANN, K.; AGATHOS, A. The Point in Polygon Problem for Arbitrary Polygons. *Computational Geometry*, v. 20, n. 3, p. 131–144, 2001.

HU, W. *et al.* A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, v. 34, n. 3, p. 334–352, 2004.

IMAGES.CV. *images.cv: Labeled Image Datasets*. [*S. l.: s. n.*], 2026. https://images.cv. Acesso em: 19 abr. 2026.

INTERNATIONAL COMMISSION ON NON-IONIZING RADIATION PROTECTION. Validity of the 1984 Interim Guidelines on Airborne Ultrasound and Gaps in the Current Knowledge. *Health Physics*, v. 127, n. 2, p. 326–347, ago. 2024. DOI: 10.1097/HP.0000000000001800.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Robots and Robotic Devices — Collaborative Robots (ISO/TS 15066)*. Geneva, Switzerland: [*s. n.*], 2016. Technical Specification.

JAMES, G. *et al. An Introduction to Statistical Learning*. [*S. l.*]: Springer, 2013.

JOCHER, G.; CHAURASIA, A.; QIU, J. *Ultralytics YOLOv8*. [*S. l.: s. n.*], 2023a. Disponível em: https://github.com/ultralytics/ultralytics.

JOCHER, G.; CHAURASIA, A.; QIU, J. *Ultralytics YOLOv8 Documentation: Tips for Best Training Results — Background Images*. [*S. l.: s. n.*], 2023b. https://docs.ultralytics.com/guides/model-training-tips/. Acesso em: 19 abr. 2026.

JULIER, S. J.; UHLMANN, J. K. Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, v. 92, n. 3, p. 401–422, 2004. DOI: 10.1109/JPROC.2003.823141.

KAGGLE. *House Rooms Image Dataset*. [*S. l.: s. n.*], 2024. https://www.kaggle.com. Acesso em: 17 mai. 2026.

KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, v. 82, n. 1, p. 35–45, 1960. DOI: 10.1115/1.3662552.

KHOSLA, A. *et al.* Novel dataset for fine-grained image categorization: Stanford dogs. *In:* FIRST Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition. [*S. l.: s. n.*], 2011. v. 2.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *In:* ADVANCES in Neural Information Processing Systems (NeurIPS). [*S. l.: s. n.*], 2012. p. 1097–1105. DOI: 10.1145/3065386.

KUHN, H. W. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, v. 2, n. 1–2, p. 83–97, 1955. DOI: 10.1002/nav.3800020109.

LI, F. *et al.* GhostConv+CA-YOLOv8n: a lightweight network for rice pest detection based on the aggregation of low-level features in real-world complex backgrounds. *Frontiers in Plant Science*, v. 16, 2025. DOI: 10.3389/fpls.2025.1620339. Disponível em: https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2025.1620339/full. Acesso em: 4 jul. 2026.

LIN, J. *et al.* Tiny Machine Learning: Progress and Futures. *IEEE Circuits and Systems Magazine*, v. 23, n. 3, p. 8–34, 2023. DOI: 10.1109/MCAS.2023.3302182.

LIN, T.-Y. *et al.* Microsoft COCO: Common Objects in Context. *In:* EUROPEAN Conference on Computer Vision (ECCV). [*S. l.: s. n.*], 2014. p. 740–755. DOI: 10.1007/978-3-319-10602-1_48.

LINDSAY, S. R. *Handbook of Applied Dog Behavior and Training, Volume 1: Adaptation and Learning*. Ames: Iowa State University Press, 2000.

LOWE, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, v. 60, n. 2, p. 91–110, 2004.

MACKINTOSH, N. J. *The Psychology of Animal Learning*. London: Academic Press, 1974.

MARTÍNEZ-BYER, S. *et al.* How Lazy Are Pet Cats Really? Using Machine Learning and Accelerometry to Get a Glimpse into the Behaviour of Privately Owned Cats in Different Households. *Applied Animal Behaviour Science*, v. 260, p. 105853, 2023.

MATHIS, A. *et al.* DeepLabCut: Markerless Pose Estimation of User-Defined Body Parts with Deep Learning. *Nature Neuroscience*, v. 21, p. 1281–1289, 2018. DOI: 10.1038/s41593-018-0209-y.

MICROPYTHON. *Overview — MicroPython latest documentation*. 2025. Disponível em: https://docs.micropython.org/. Acesso em: 11 nov. 2025.

MICROSOFT. *ONNX Runtime — Official Documentation*. 2025. Disponível em: https://onnxruntime.ai/docs/. Acesso em: 11 nov. 2025.

MUNKRES, J. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, v. 5, n. 1, p. 32–38, 1957. DOI: 10.1137/0105003.

NVIDIA. *Jetson Orin Nano Super Developer Kit — Product Page*. 2025. Disponível em: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/nano-super-developer-kit/. Acesso em: 11 nov. 2025.

O'ROURKE, J. *Computational Geometry in C*. 2. ed. Cambridge: Cambridge University Press, 1998.

OGATA, K. *Modern Control Engineering*. 5. ed. [*S. l.*]: Prentice Hall, 2010.

OPENCV. *OpenCV 4.x Documentation*. 2025. Disponível em: https://docs.opencv.org/4.x/. Acesso em: 11 nov. 2025.

OTSU, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, n. 1, p. 62–66, 1979.

OVERALL, K. L. *Manual of Clinical Behavioral Medicine for Dogs and Cats*. St. Louis: Elsevier, 2013.

PAN, S. J.; YANG, Q. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, v. 22, n. 10, p. 1345–1359, 2009.

PAVLOV, I. P. *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*. London: Oxford University Press, 1927.

PETSAFE. *Smart Feed Automatic Dog and Cat Feeder (2nd Gen)*. 2025. Disponível em: https://www.petsafe.com/product/smart-feed/. Acesso em: 11 nov. 2025.

PICCIONE, G. *et al.* Daily rhythms in food intake and locomotor activity in a colony of domestic cats. *Journal of Veterinary Behavior*, v. 8, n. 4, p. 260–264, 2013.

POWERS, D. M. W. Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. *Journal of Machine Learning Technologies*, v. 2, n. 1, p. 37–63, 2011. Disponível em: https://arxiv.org/abs/2010.16061. Acesso em: 3 nov. 2025.

PRECHELT, L. Early Stopping - But When? *In:* NEURAL Networks: Tricks of the Trade. [*S. l.*]: Springer, 1998.

PYTORCH. *PyTorch Documentation (stable)*. 2025. Disponível em: https://pytorch.org/docs/stable/. Acesso em: 11 nov. 2025.

RASPBERRY PI LTD. *Raspberry Pi 4 Model B — Product Page*. 2025. Disponível em: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/. Acesso em: 11 nov. 2025.

REDMON, J.; DIVVALA, S. *et al.* You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 779–788, 2016.

REDMON, J.; FARHADI, A. YOLOv3: An Incremental Improvement. *In:* ARXIV preprint arXiv:1804.02767. [*S. l.: s. n.*], 2018.

REN, S. *et al.* Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *In:* ADVANCES in Neural Information Processing Systems (NeurIPS). [*S. l.: s. n.*], 2015. p. 91–99.

SANTOS, R. C. C. D. M.; COELHO, M.; OLIVEIRA, R. Real-time Object Detection Performance Analysis Using YOLOv7 on Edge Devices. *IEEE Latin America Transactions*, v. 22, n. 10, p. 799–805, 2024. DOI: 10.1109/TLA.2024.10705971.

SCHEIFELE, P. *et al.* Effect of kennel noise on hearing in dogs. *American Journal of Veterinary Research*, v. 73, n. 4, p. 482–489, 2012. DOI: 10.2460/ajvr.73.4.482.

SCHILLING, N.; CARRIER, D. R. Function of the epaxial muscles in walking, trotting and galloping dogs: implications for the evolution of epaxial muscle function in tetrapods. *Journal of Experimental Biology*, v. 211, n. 1, p. 138–149, 2008. DOI: 10.1242/jeb.011213.

SCHIZAS, N. *et al.* TinyML for Ultra-Low Power AI and Large Scale IoT Deployments: A Systematic Review. *Future Internet*, v. 14, n. 12, p. 363, 2022. DOI: 10.3390/fi14120363.

SHAFIQUE MUHAMMAD, E. A. An Overview of Deep Learning On-Edge: Architecture, Optimizations, and Trends. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.

SHI, W. *et al.* Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, v. 3, n. 5, p. 637–646, 2018.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, v. 6, n. 1, p. 1–48, 2019.

SIMONYAN, K.; ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICLR)*, 2015. arXiv:1409.1556.

SKINNER, B. F. *Science and Human Behavior*. New York: Macmillan, 1953.

SKINNER, B. F. *The Behavior of Organisms: An Experimental Analysis*. New York: Appleton-Century, 1938.

SKINNER, B. F. Operant Behavior. *American Psychologist*, v. 21, n. 5, p. 503–515, 1966. DOI: 10.1037/h0023465.

STM MICROELECTRONICS. *VL53L0X Time-of-Flight Ranging Sensor*. Datasheet. [*S. l.*], 2024. Disponível em: https://www.st.com/resource/en/datasheet/vl53l0x.pdf. Acesso em: 11 nov. 2025.

SUTHERLAND, I. E.; HODGMAN, G. W. Reentrant Polygon Clipping. *In:* 1. COMMUNICATIONS of the ACM. [*S. l.: s. n.*], 1974. v. 17, p. 32–42.

SZELISKI, R. *Computer Vision: Algorithms and Applications*. 2. ed. Cham: Springer, 2022.

TENSORFLOW. *TensorFlow — Core Guide and API*. 2025a. Disponível em: https://www.tensorflow.org/guide. Acesso em: 11 nov. 2025.

TENSORFLOW. *TensorFlow Lite — Guide*. 2025b. Disponível em: https://www.tensorflow.org/lite/guide. Acesso em: 11 nov. 2025.

TENSORFLOW. *TensorFlow Lite for Microcontrollers*. Acessado em: 11 dez. 2025. 2021. Disponível em: https://www.tensorflow.org/lite/microcontrollers.

THE PET COLLECTIVE. *These cats are too funny! New cat videos 2025*. [*S. l.: s. n.*], abr. 2025. Jukin Media. 1 vídeo. Disponível em: https://youtu.be/3URtTIdnXIk.

ULTRALYTICS. *Ultralytics YOLO — Documentation*. 2025. Disponível em: https://docs.ultralytics.com/. Acesso em: 11 nov. 2025.

VILLANI, V. *et al.* Survey on Human–Robot Collaboration in Industrial Settings: Safety, Intuitive Interfaces and Applications. *Mechatronics*, v. 55, p. 248–266, 2018. DOI: 10.1016/j.mechatronics.2018.02.009.

VIVEKANANDA, H. P. S. *et al.* Integration of Aerial Object Detection System with Flight Controller. *In:* 2025 IEEE International Circuit and System Symposium (ICSyS). Kuala Lumpur, Malaysia: IEEE, 2025. Date Added to IEEE Xplore: 19 Feb. 2026. DOI: 10.1109/ICSyS65149.2025.11379891.

VOSS, K. *et al.* Kinetic and temporospatial gait parameters in a heterogeneous group of dogs. *BMC Veterinary Research*, v. 16, n. 1, p. 1–11, 2020.

WELCH, G.; BISHOP, G. *An Introduction to the Kalman Filter*. [*S. l.*], 2001. Revisado em 2006.

WORLD ANIMAL PROTECTION. *The Global State of Pet Ownership 2022*. [*S. l.: s. n.*], 2022. https://www.worldanimalprotection.org/latest/blogs/global-state-of-pet-ownership/. Acesso em: 5 out. 2025.

YOSINSKI, J. *et al.* How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

ZAGITOV, A. *et al.* Comparative analysis of neural network models performance on low-power devices for a real-time object detection task. *Computer Optics*, v. 48, n. 2, p. 242–252, 2024. DOI: 10.18287/2412-6179-CO-1343.