



DESENVOLVIMENTO DE ALGORITMO DE NAVEGAÇÃO E VISÃO COMPUTACIONAL PARA COMPETIÇÃO DE DRONES

Herick Bento Lourenço | herick.b2005@aluno.ifsc.edu.br¹
Gabriel Beu Nogueira de Macedo | gabriel.macedo@ifsc.edu.br²

Trabalho de Pesquisa
DT 4 - Inovação Tecnológica, Processos Produtivos e Futuro das Tecnologias

Resumo:

Este trabalho descreve o desenvolvimento de um sistema de navegação autônoma e de visão computacional para um veículo aéreo não tripulado (VANT), projetado para atender aos desafios da competição EletroQuad, promovida pela SAE Brasil. O objetivo é fazer o drone decolar, procurar bases, identificar um marcador ArUco embutido em uma forma geométrica na base e, a partir da leitura do identificador (ID) ArUco, realizar o pouso em uma base específica. A metodologia englobou a modelagem da arquitetura de *software* via diagramas de blocos e o desenvolvimento do controle de trajetória em padrão zigue-zague utilizando o *Robot Operating System* (ROS). Para a etapa de identificação de objetos, adotou-se a rede neural *You Only Look Once* (YOLO), treinada com um *dataset* customizado de imagens reais e anotações normalizadas, visando a detecção em tempo real de formas geométricas e numerais em *hardware* embarcado (Raspberry Pi). Previamente à execução física, o sistema foi integralmente validado em um ambiente de simulação construído no *software* Gazebo. Os resultados demonstraram o êxito da integração dos módulos de controle e visão.

Palavras-chave: visão computacional; YOLO; navegação autônoma.

Introdução

Nos últimos anos, o uso de VANTs, popularmente conhecidos como drones, tem se expandido para aplicações como inspeção de redes elétricas, agricultura de precisão e operações de busca e resgate. Para que essas aeronaves executem tarefas sem intervenção humana direta, são necessários sistemas de navegação autônoma e inteligência artificial embarcada.

Nesse contexto, competições universitárias estimulam a pesquisa e a inovação tecnológica. A competição EletroQuad, promovida pela SAE Brasil, desafia equipes a

¹Estudante do curso de Engenharia Eletrônica do IFSC - Câmpus Florianópolis.

² Servidor do Departamento Acadêmico de Eletrotécnica (DAE) do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

desenvolver drones capazes de integrar controle de voo, visão computacional e tomada de decisão em tempo real.

Na missão “Bouncing 2.0”, o drone deve localizar um marcador ArUco posicionado dentro de uma figura geométrica e, a partir do ID identificado, aplicar regras de divisibilidade para determinar autonomamente a base correta de pouso. Todo o processamento deve ocorrer em *hardware* embarcado, com recursos computacionais limitados.

Diante desse cenário, o trabalho tem como objetivo descrever o desenvolvimento e a incorporação de um sistema de navegação autônoma e reconhecimento de padrões para cumprir a missão proposta. A relevância do projeto está na aplicação prática de conceitos de robótica móvel, redes neurais e simulação computacional voltados à competição EletroQuad.

Fundamentação teórica

O desenvolvimento de veículos autônomos exige a integração de controle, percepção do ambiente e inteligência artificial. Para isso, utiliza-se o sistema ROS (ROS, 2026), um *middleware* de código aberto que fornece bibliotecas e comunicação modular, facilitando a implementação de algoritmos. O ROS integra-se nativamente ao simulador Gazebo (GAZEBO, 2026), etapa indispensável para validar lógicas em um ambiente seguro, mitigando danos ao *hardware* na transição para o mundo real.

Na percepção do ambiente, Redes Neurais Convolucionais (CNNs) representam o estado da arte para detecção de objetos em tempo real. O modelo YOLO destaca-se por resolver a detecção como um único problema de regressão matemática, mapeando pixels para coordenadas de *bounding boxes* e classes simultaneamente. A execução de redes neurais convolucionais em VANTs impõe severas restrições de processamento, peso e consumo energético ao *hardware* embarcado, como a Raspberry Pi 5. A ausência de uma GPU dedicada nesses dispositivos cria um gargalo computacional para inferências em tempo real. Nesse cenário, a arquitetura YOLOv11n (Nano) torna-se fundamental, pois foi projetada para ambientes de recursos limitados, equilibrando uma taxa aceitável de quadros por segundo (FPS) com baixo custo computacional, evitando o superaquecimento e o esgotamento da memória RAM do sistema (DUARTE; ALMEIDA,

2025).

Por fim, através da biblioteca OpenCV (OPENCV, 2026), o sistema detecta os marcadores com alta confiabilidade, mesmo sob variações de iluminação, e extrai seu identificador único.

Procedimentos metodológicos

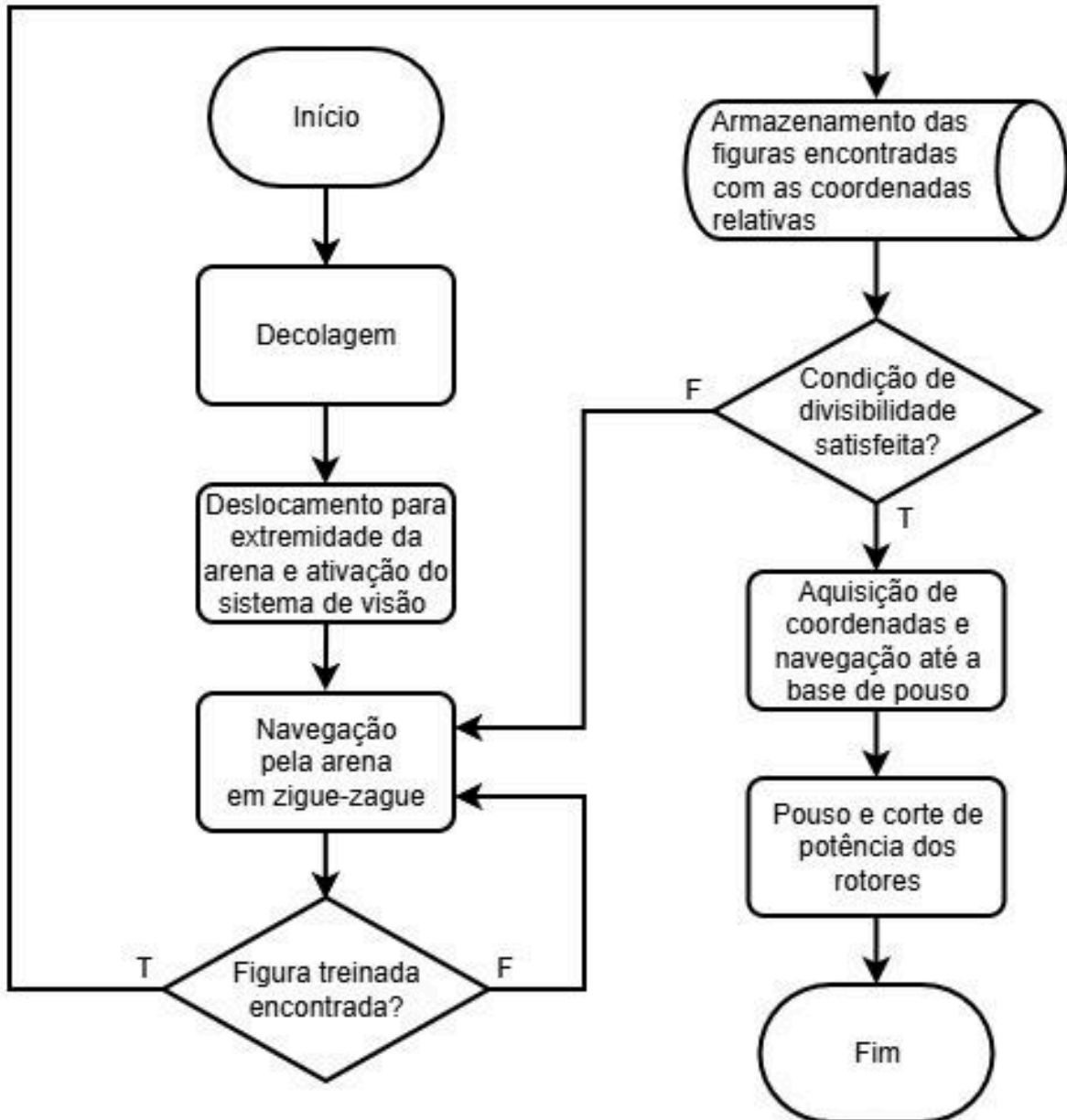
A metodologia consiste no desenvolvimento dos códigos de controle da missão em Python, do treinamento do modelo de detecção de objetos com YOLOv11n e testes de validação do sistema. O sistema será validado primeiramente no ambiente de simulação Gazebo. Posteriormente, com êxito obtido em vários cenários de simulação, o código será testado no drone real.

Para o desenvolvimento do algoritmo, foi elaborado o diagrama de blocos da Figura 1, com o processamento de imagens e envio de comandos para o drone. A partir do diagrama de blocos, o desenvolvimento do *software* foi realizado de forma modular, com a implementação gradual de cada componente do sistema.

A primeira etapa do desenvolvimento técnico focou no algoritmo de busca. Optou-se por um padrão de navegação em zigue-zague para garantir a cobertura completa da arena. O controle de movimentação foi implementado utilizando o *framework* ROS, por meio de uma função de deslocamento que calcula a distância euclidiana entre a posição atual e a posição desejada. Cada movimentação do drone é feita com um novo comando.

A segunda etapa concentrou-se no sistema de visão computacional para o reconhecimento das formas geométricas (estrela, triângulo e hexágono) e dos numerais associados. A construção do *dataset* para o treinamento do modelo envolveu a criação de bases de pouso conforme as especificações da SAE Brasil. A partir da confecção, foram feitas as imagens para o *dataset* com a câmera do drone em ambiente de laboratório, para simular as condições reais de iluminação. A base de dados customizada reuniu 329 imagens (262 para treino e 67 para validação), rotuladas com coordenadas normalizadas e classes em formato .txt. O treinamento do modelo utilizou aceleração por *hardware* (CUDA) ao longo de 200 épocas com lotes de 16 imagens.

Figura 1 – Diagrama de blocos para a missão “Bouncing 2.0” da EletroQuad.



Fonte: Produção própria (2026).

As técnicas de *data augmentation* foram otimizadas para o reconhecimento das formas e numerais: espelhamentos e distorções foram desativados para evitar a inversão geométrica, aplicando-se apenas rotações ($\pm 10^\circ$) e variações de brilho e escala para simular a dinâmica de voo, sombreamento e altitude. O processo gerou o arquivo de pesos ('best.pt'), capacitando o sistema a classificar formas e números simultaneamente (Figura 2).

Figura 2 – Classificação da base pelo modelo treinado e a respectiva confiança.

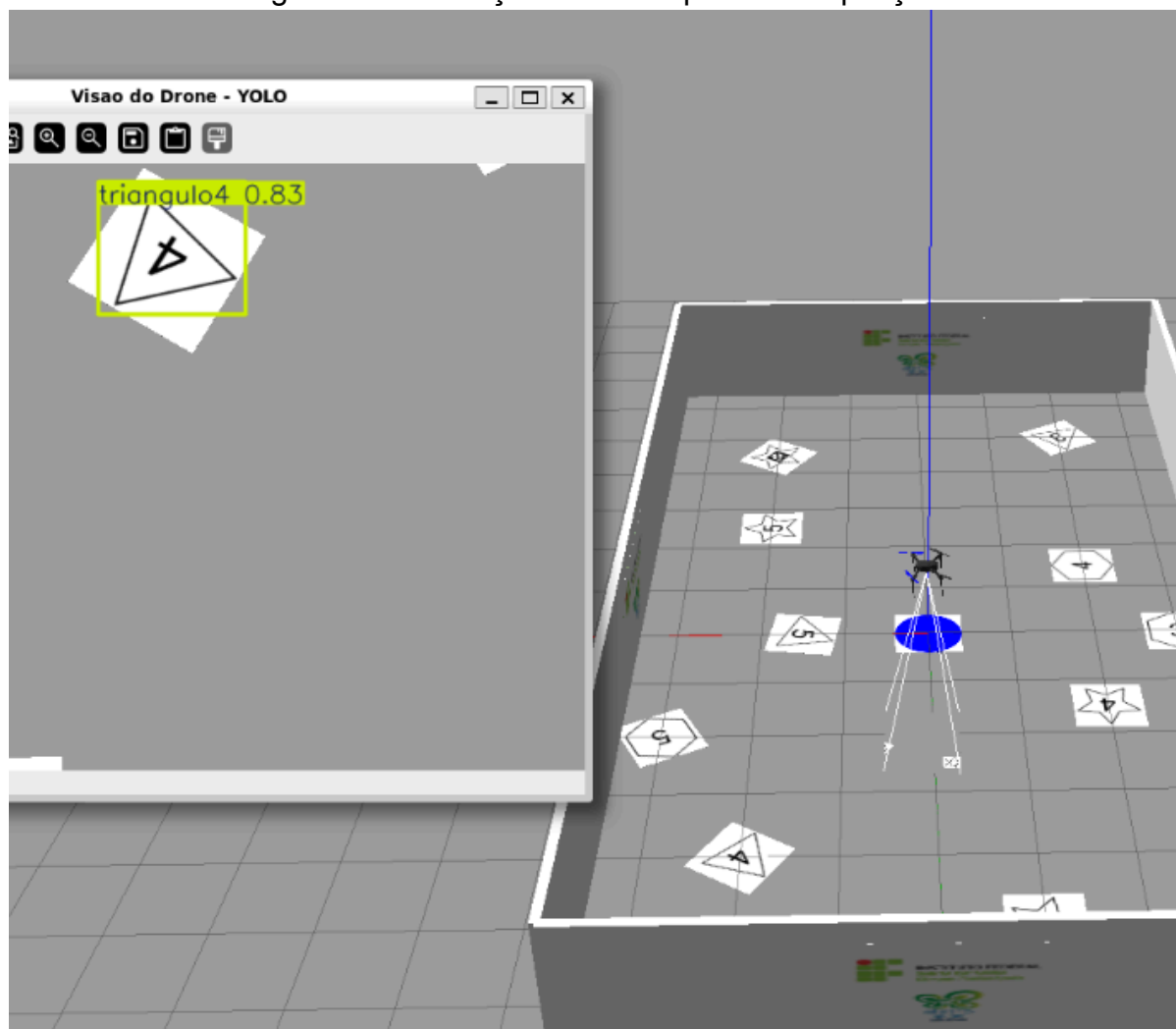


Fonte: Produção própria (2026).

Paralelamente, foi construído um ambiente de simulação no *software* Gazebo, executado em arquitetura *Windows Subsystem for Linux 2 (WSL2)*. A arena foi modelada para replicar as dimensões e desafios da competição, permitindo a validação dos algoritmos de reconhecimento e a lógica de decisão, sem riscos ao equipamento real. Embora o ambiente simulado apresente divergências do mundo real, o uso do Gazebo otimiza o tempo de desenvolvimento ao permitir a validação prévia do comportamento do drone.

Por fim, a integração dos módulos permite a realização da missão simulada, conforme mostrado na Figura 3. Dessa forma, é possível testar a movimentação para procurar os objetos, o modelo de reconhecimento de objetos, a lógica aritmética para decisão de qual base pousar, e a identificação e pouso na superfície desejada. Todo o processo metodológico evidencia que o sucesso em missões de robótica aérea depende da organização rigorosa, simulação exaustiva e otimização de modelos de inteligência artificial para o *hardware* disponível.

Figura 3 – Simulação de arena para a competição.



Fonte: Produção própria (2026).

Resultados e discussões

Os resultados foram obtidos utilizando o simulador Gazebo. O algoritmo de varredura em zigue-zague foi totalmente validado. Na visão computacional, o modelo YOLOv11n obteve excelente desempenho quantitativo, operando a uma taxa de processamento de 3 Hz (3 quadros por segundo), o que se mostrou adequado para a dinâmica de navegação e tomada de decisão. As métricas de validação registraram precisão média (mAP50) de 0,99 e mAP50-95 de 0,81, um indicador rigoroso que atesta o exato alinhamento das caixas delimitadoras com as bordas reais dos objetos. O modelo



operou com tempo de inferência inferior a 330 ms na CPU (3 FPS) e taxa de erro por falsos positivos menor que 2%. Esses índices traduzem uma acurácia quase perfeita na classificação das formas e numerais, além de alta precisão na localização espacial.

Durante a execução simulada, a rede manteve níveis de confiança consistentes em todas as detecções. A lógica de missão funcionou conforme o esperado: o sistema identificou o ArUco, decodificou seu ID, aplicou a regra de divisibilidade e localizou a superfície de pouso correta, ignorando com sucesso as demais. O registro das coordenadas espaciais de cada base localizada será aprimorado nas próximas etapas. Esse procedimento permitirá que o drone se desloque diretamente para o alvo correto após a leitura e decodificação do marcador fiducial, otimizando o tempo total de execução da missão. Embora a simulação reduza os riscos do desenvolvimento, reconhece-se que a transição para a operação física apresenta limitações inerentes ao ambiente não controlado. Fatores externos como variações abruptas de iluminação natural, desfoque de movimento da câmera (*motion blur*) e possíveis oscilações na taxa de quadros (FPS) devido ao aquecimento do *hardware* embarcado deverão ser rigorosamente monitorados e mitigados nos ensaios práticos.

Considerações finais

Em suma, os objetivos propostos para a realização desta missão foram plenamente alcançados. O uso de redes neurais na identificação de imagens personalizadas validou a precisão do sistema, demonstrando ser uma tecnologia com forte potencial de aplicação na indústria brasileira. A execução deste projeto evidencia que a participação em competições tecnológicas e o engajamento em pesquisa aplicada consolidam a inovação e fortalecem a formação acadêmica e profissional na área.

Referência ao fomento recebido

Este trabalho foi viabilizado com apoio do Câmpus Florianópolis do IFSC, com a viabilização de espaço físico para as atividades do Laboratório de Drones e destaque para o DAE (Departamento Acadêmico de Eletrotécnica) que fornece a infraestrutura complementar do laboratório. Registra-se também o apoio da FEESC (Fundação



Stemmer para Pesquisa, Desenvolvimento e Inovação), por meio do edital “Conexões para Inovar”, que contemplou a equipe de drones do Câmpus Florianópolis nos anos de 2024 a 2026.

Referências

ROS. Robot Operating System. 2026. Disponível em: <https://www.ros.org/>. Acesso em: 22 maio 2026.

GAZEBO. Gazebo Simulator. 2026. Disponível em: <https://gazebo.org/>. Acesso em: 22 maio 2026.

DUARTE, Luize Cunha; ALMEIDA, Paulo Ricardo Lisboa de. Aplicação da Arquitetura YOLOv11 Nano na Classificação de Raças de Cães. 2025. Disponível em: <https://periodicos.univali.br/index.php/acotb/article/view/21110/12178>. Acesso em: 28 abr. 2025.

OPENCV. Open Source Computer Vision Library. 2026. Disponível em: <https://opencv.org/>. Acesso em: 22 maio 2026.