

# Delta High Team Description Paper

1<sup>st</sup> Gabriel Beu Nogueira de Macedo  
*Professor do departamento de  
Eletrotécnica - IFSC*  
Florianópolis, Brasil  
gabriel.macedo@ifsc.edu.br

2<sup>nd</sup> Leandro de Medeiros Sebastião  
*Professor do departamento de  
Eletrotécnica - IFSC*  
Florianópolis, Brasil  
leandro.medeiros@ifsc.edu.br

3<sup>rd</sup> João Pedro de Araújo Duarte  
*Aluno de Engenharia Eletrônica - IFSC*  
Florianópolis Brasil  
joao.pad@aluno.ifsc.edu.br

4<sup>th</sup> Jean Monteiro Azevedo Chaves  
*Aluno de Engenharia Elétrica - IFSC*  
Florianópolis, Brasil  
jean.ac18@aluno.ifsc.edu.br

5<sup>th</sup> Antonio Augusto G. de Oliveira  
*Aluno do Tecnólogo em Sistemas de  
Energia - IFSC*  
Florianópolis, Brasil  
antonio.ag@aluno.ifsc.edu.br

6<sup>th</sup> Enzo Renan Moreira Tralli  
*Aluno de Engenharia Mecatrônica - IFSC*  
Florianópolis Brasil  
enzo.rmt@aluno.ifsc.edu.br

7<sup>th</sup> Maria Luiza dos Santos  
*Aluno do Técnico em Eletrônica - IFSC*  
Florianópolis Brasil  
maria.ls2005@aluno.ifsc.edu.br

8<sup>th</sup> Fabrício Rodrigues de Santana  
*Aluno de Engenharia Eletrônica - IFSC*  
Florianópolis Brasil  
fabricio.r11@aluno.ifsc.edu.br

9<sup>th</sup> Ariel Fernandes  
*Aluno de Engenharia Eletrônica - IFSC*  
Florianópolis Brasil  
ariel.f@aluno.ifsc.edu.br

10<sup>th</sup> Diana Macedo Rodrigues  
*Aluno de Engenharia Eletrônica - IFSC*  
Florianópolis Brasil  
diana.m09@aluno.ifsc.edu.br

11<sup>th</sup> Luiza Yasmim Feijó da Silva  
*Aluno do Técnico em Eletrônica - IFSC*  
Florianópolis Brasil  
luiza.yf@aluno.ifsc.edu.br

12<sup>th</sup> Marina Espindola Bampi e Silva  
*Aluno de Engenharia Eletrônica - IFSC*  
Florianópolis Brasil  
marina.ebs@aluno.ifsc.edu.br

13<sup>th</sup> Beatriz Spricigo Magalhães  
*Aluno de Engenharia Eletrônica - IFSC*  
Florianópolis Brasil  
beatriz.m18@aluno.ifsc.edu.br

**Abstract** — This article presents an overview of Delta High Team's preparation for the Brazilian Robotics Competition at the RoboCup Brazil 2024 - Flying Robot Trial League. With this, the team expects to contribute to future participants on making an autonomous flying vehicle. This year's TDP mainly focuses on describing the new Q260 prototype and the hardware and software architecture. Furthermore, the development of a mapping and localization system is explored.

**Index Terms** — RoboCup Brazil, Autonomous Drone Architecture, Mapping and Localization System.

LINK to the Video: <https://youtu.be/VKMqZ2ehRco>

## I. OVERVIEW

A Delta High, equipe de competição do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina (IFSC) do campus Florianópolis, tem trabalhado em melhorias significativas das aeronaves autônomas, desenvolvidas com manufatura aditiva e projetos *open hardware*. Neste artigo, a equipe descreve as principais inovações que estão sendo aplicadas para a Competição Brasileira de Robótica 2024, com

melhorias do protótipo Q450 e o desenvolvimento de um novo conceito com o drone Q260: um protótipo menor, mais leve e com proteção de hélice para operação segura em interações com humanos.

Também é abordado o sistema de mapeamento e localização, que começou a ser explorado pela equipe neste ano, e o sistema de detecção de bases de pouso por visão computacional.

### A. Inovação

#### 1) Inovações no protótipo Q450:

- **Melhorias mecânicas:** Com o objetivo de otimizar a aeronave utilizada em 2023, a equipe propôs nova versão do trem de pouso, redimensionou as *center-plates* e desenvolveu uma nova proposta para instalação dos dispositivos embarcados na aeronave.
- **Novos componentes:** Para ampliar a capacidade de mapeamento e localização, foram instalados novos componentes como o lidar rotativo e a câmera de profundidade.

## 2) Novo protótipo Q260:

- **Frame mecânico:** Foi desenvolvido um novo frame mecânico para atender aos requisitos da fase 4 da competição, sendo eles o tamanho máximo de 330mm entre eixos de motores opostos, peso máximo de decolagem de 1 quilograma e uso de proteção de hélice.
- **ESC 4 em 1:** Para otimizar o peso e volume, foi utilizado um controlador de velocidade único para os 4 motores, com capacidade de 50A.
- **Motores Racer:** Para reduzir o tamanho das hélices e viabilizar o frame reduzido, foram utilizados motores de modelo RS2205 de 2300KV, geralmente usados em drones racer.

## 3) Sistema de detecção do ambiente:

- **Lidar Rotativo:** Foi instalado um lidar rotativo de 360 graus na parte superior do drone, com propósito de detecção de obstáculos. Este componente permitirá pesquisas com algoritmos de mapeamento e localização.

## II. SISTEMA ELÉTRICO E MECÂNICO

Este ano, a equipe trouxe vários avanços mecânicos e eletrônicos no drone robô Q450, como melhorias no frame e no trem de pouso, bem como a instalação do lidar rotativo e câmera de profundidade. Também foi desenvolvido um novo protótipo para participar da fase 4, aqui denominado drone robô Q260, pois possui 260mm entre eixos de motores opostos.

### A. Drone Robô Q450

Para a melhoria do trem de pouso, a equipe utilizou uma nova metodologia de montagem, substituindo a utilização de pinos para fixar no braço do motor por um encaixe de pressão, mostrada na Figura 1. Essa abordagem resultou na facilidade para uma eventual substituição de braço ou do trem de pouso.

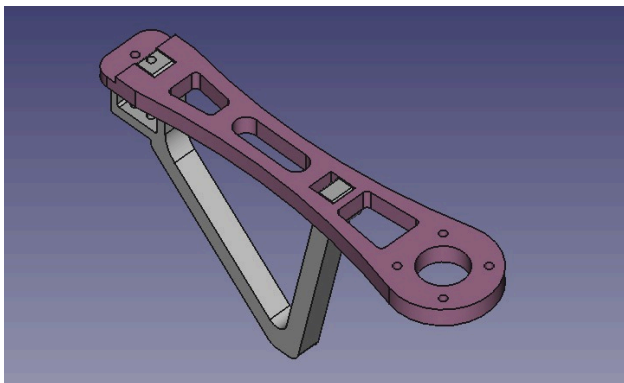


Figura 1. Novo trem de pouso com sistema de encaixe.

O redimensionamento das *center plates* foi necessário, tendo em vista a aquisição de novos componentes para o drone. Desta forma, a alteração foi realizada para integrar os novos suportes para a Raspberry Pi 4, lidar LD19 e a câmera de profundidade Intel Realsense D435i. O *layout* atual do drone pode ser visualizado na Figura 2. Na versão de 2023 do protótipo, só era possível utilizar uma única câmera, ou fixada para frente ou para baixo, o que dificultava o *setup* para testes e para a competição.

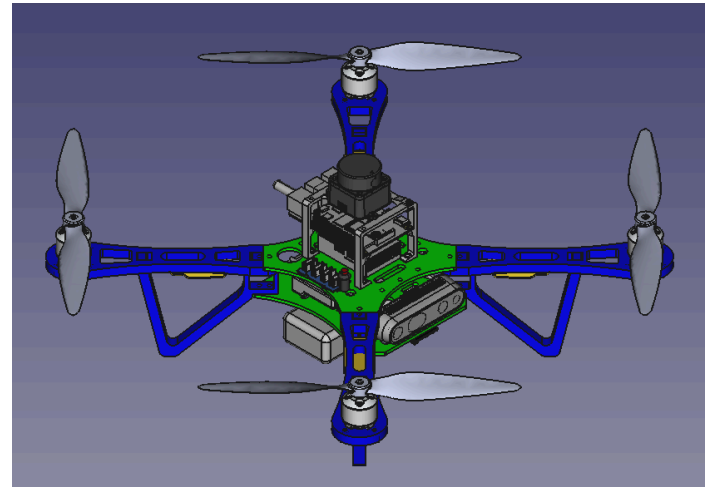


Figura 2. Modelagem 3D do drone Q450 no software FreeCAD.

Com a instalação dos novos dispositivos, obteve-se um *setup* mais robusto e com possibilidade de desenvolver metodologias avançadas para mapeamento e localização em ambientes desconhecidos. A Tabela I compara as duas versões do robô.

TABELA I  
ESPECIFICAÇÕES DO DRONE ROBÔ Q450

Versão do Robô	2023	2024
Dimensão	450mm	450mm
Peso total	1,30kg	1,40kg
Motores	2212 1400KV	2212 1400KV
Hélices	8"	8"
Controladora de voo	Pixhawk 2.4.8	Pixhawk 2.4.8
Câmera frontal	RaspCam 5 MP	Intel Realsense D435i
Câmera Inferior	RaspCam 5 MP	RaspCam 5 MP
Odometria	HereFlow	HereFlow / Lidar LD19

O diagrama das conexões elétricas pode ser observado na Figura 3. Neste diagrama foi utilizado um controlador de velocidade para 4 motores (ESC 4 in 1) o qual será utilizado no drone Q260. Como a equipe só tinha disponível uma unidade deste componente, no Q450 foram utilizados 4 controladores independentes, um para cada motor. Também deve-se ressaltar que a alimentação da Raspberry Pi 4 é proveniente do UBEC (*Universal Battery Eliminator Circuit*), conectado na saída do *Power Module*, para garantir energia suficiente para sua operação. Como a Pixhawk é alimentada por outra fonte (interna do *Power Module*), não se deve conectar os VCCs (5V) entre estes dispositivos, principalmente no cabo de comunicação UART, na porta de telemetria da Pixhawk.

O sensor HereFlow é utilizado para medir a velocidade XY da aeronave, permitindo ao controlador de voo estimar a posição atual e viabilizando o voo autônomo sem GPS. Ademais, um lidar unidirecional é utilizado para medição precisa de altura relativa, essencial para o funcionamento do HereFlow. A documentação destes dispositivos aplicados em conjunto com a Pixhawk pode ser obtida em [1] e [2]. Algumas das configurações e calibrações já foram contempladas no TDP de 2023 da equipe, conforme [3].

A arquitetura de software utilizada é apresentada na Figura 4. Nesta concepção, a Raspberry Pi 4 é utilizada como *companion computer* (computador de missão) e todo o processamento de visão computacional e inteligência artificial (IA) é feito

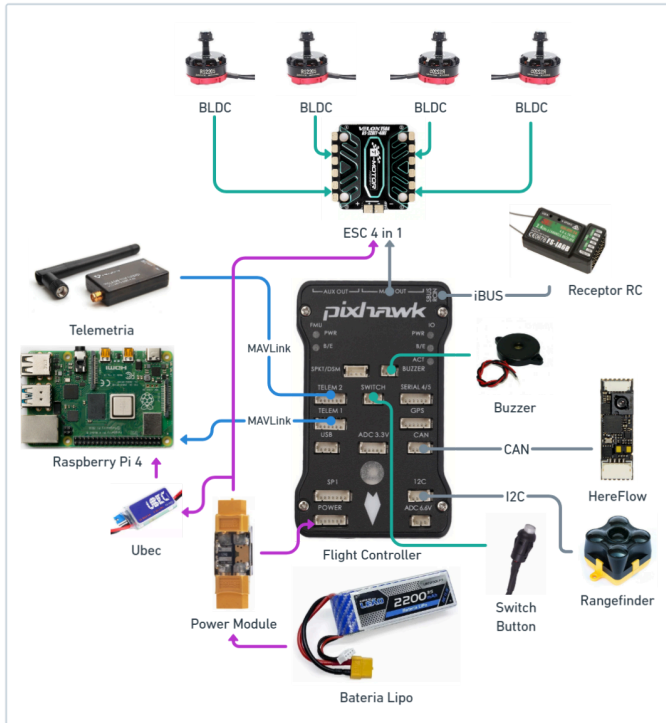


Figura 3. Arquitetura de hardware embarcado.

embarcado na aeronave. Os dispositivos periféricos para navegação são conectados à Raspberry Pi, tais como lidar rotativo e câmeras. O sistema foi desenvolvido para ser compatível com o ROS/ROS2 ou também com aplicações concorrentes em *threads* separadas. Dessa forma, garante maior versatilidade para o trabalho da equipe e é possível isolar os códigos por funcionalidade, seja tratando a imagem obtida ou processando inferências em um modelo treinado.

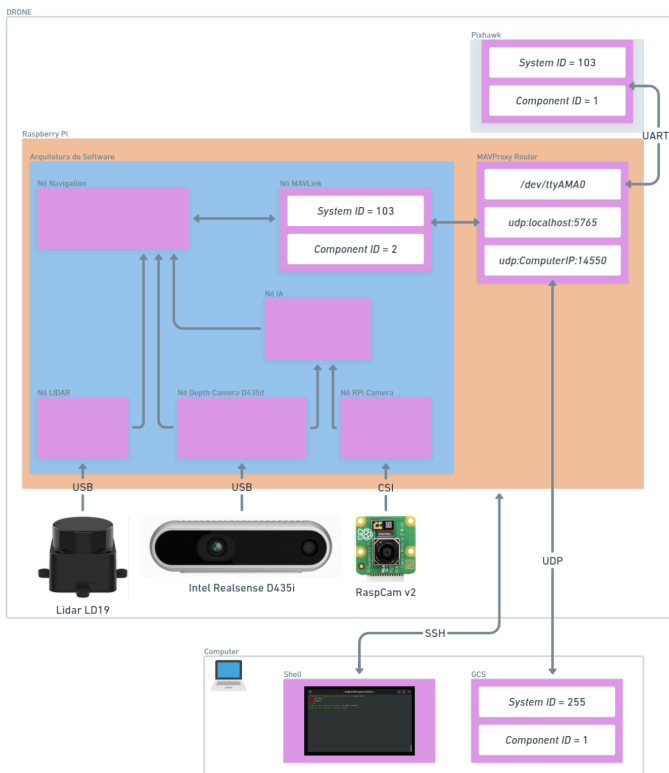


Figura 4. Arquitetura de software utilizada.

Os comandos são enviados para a aeronave pelo protocolo MAVLink [4], através da API (Interface de Programação de Aplicação) Pymavlink [5]. Para se ter acesso à telemetria pela estação de solo (*Ground Control Station - GCS*) é utilizado o *software* MAVProxy como hub, recebendo todas as mensagens MAVLink e transmitindo para cada dispositivo de acordo com o *System ID*. Para tanto, é importante que cada sistema e cada componente dentro do sistema esteja corretamente configurado. Neste caso foi utilizado o sistema 103 para o drone, com componente 1 sendo a Pixhawk e 2 a Raspberry Pi 4, e a GCS no computador do operador com sistema 255.

### B. Drone Robô Q260

Com o propósito de realizar a fase 4 da competição (drone controlado por gestos do operador humano), a equipe desenvolveu um drone Q260 para atender as restrições de tamanho, peso e segurança. Dessa forma, este drone possui distância entre eixos de motores de 260mm. Além disso, as proteções de hélices também foram desenvolvidas para atender ao regulamento.

Para a fabricação desta aeronave, o projeto mecânico foi dividido em quatro partes distintas, viabilizando a produção e apresentando boa resistência estrutural. Desse modo, a primeira parte compreende na *center plate* para sustentação do sistema propulsivo e alocação dos demais dispositivos. Em sequência, foi elaborado um trem de pouso para a estrutura, de forma que a bateria utilizada fosse fixada ao mesmo por fitas de velcro. Para embarcar a *companion computer* (Raspberry Pi 4) e alocar demais componentes, foi desenvolvido um suporte central com dois níveis. Por fim, foi elaborada uma proteção de hélice com sistema de encaixe por pressão. A Figura 5 ilustra o modelo em 3D do drone Q260 desenvolvido pela equipe.

Com o desenvolvimento deste drone, a equipe conseguiu otimizar o projeto, atingindo um MTOW (*Maximum Take-Off Weight*, ou peso máximo de decolagem) de aproximadamente 930 gramas. A Tabela II resume as características técnicas obtidas com o protótipo.

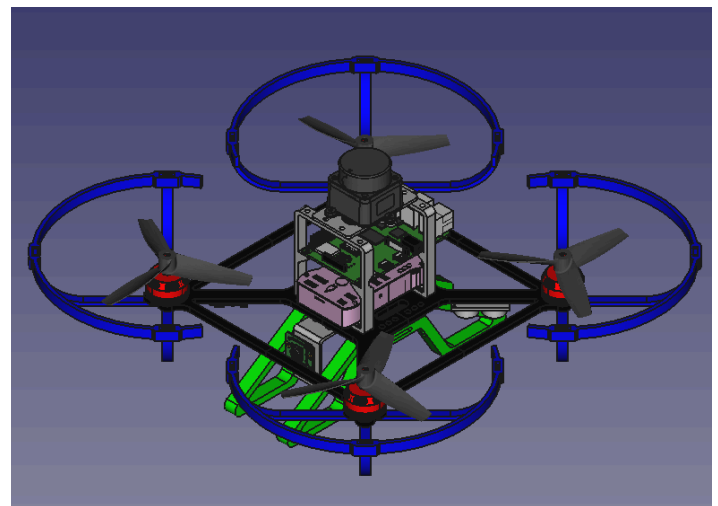


Figura 5. Modelagem 3D do drone Q260 no *software* FreeCAD.

TABELA II  
ESPECIFICAÇÕES DO DRONE ROBÔ Q260

Versão do Robô	2024
Dimensão	260mm
Peso total	930g
Motores	RS2205 2300KV
Hélices	Tripá 6"
Controladora de voo	Pixhawk 2.4.8
Câmera frontal	RaspCam v2 8 MP
Câmera Inferior	-
Odometria	HereFlow / Lidar LD19

Para a fabricação das peças descritas, foi utilizada uma impressora 3D, modelo GTMAX3D CORE A1V2. O material escolhido para a produção foi o polímero ABS, devido ao seu desempenho satisfatório em termos de peso e resistência estrutural. Esse material garante que as peças fabricadas sejam leves e robustas, atendendo aos requisitos de durabilidade necessários para a aplicação. O protótipo final obtido pela equipe é apresentado na Figura 6. Os arquivos de cada peça para impressão podem ser encontrados no GitHub da equipe [6].

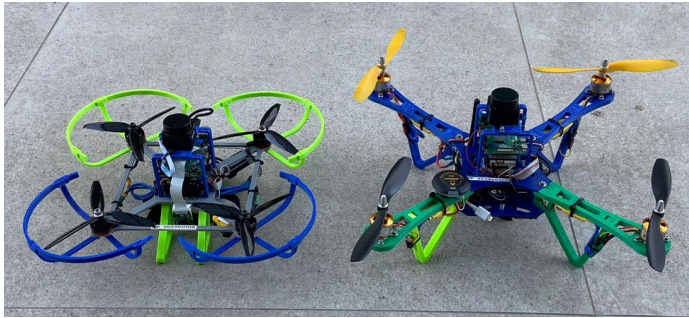


Figura 6. Drones da equipe em sua versão final.

### III. SISTEMA DE DETECÇÃO E LOCALIZAÇÃO DO AMBIENTE

Uma etapa crucial para a viabilização de operações com drones e sua aplicação em missões subsequentes é a detecção e localização do ambiente em tempo real. Para esta tarefa, um dos principais sensores utilizados é o lidar rotativo DTOF LIDAR LD19. Este sensor é fundamental para a navegação precisa e segura do drone, fornecendo dados essenciais sobre o ambiente ao seu redor.

A Tabela III apresenta as principais características de medição deste sensor, enquanto a Figura 7 ilustra sua aparência e estrutura.

TABELA III  
ESPECIFICAÇÕES DO LiDAR

Parâmetro	Valor
Faixa de medição linear	0,02~12m
Precisão de medição	10mm (300mm < medição <= 12000mm)
Resolução angular	$\leq 1^\circ$
Faixa de medição angular	360°
Frequência de varredura	5~13Hz
Peso	47g



Figura 7. LiDAR DTOF LD19.

O DTOF LIDAR LD19 é um sensor rotativo, permitindo a obtenção de leituras de distância entre um anteparo e o sensor em incrementos menores que  $1^\circ$ . Este dispositivo fornece duas informações fundamentais: a distância e o ângulo. Com essas informações, é possível gerar um gráfico que apresenta um conjunto de pontos, facilitando a identificação de obstáculos como paredes e colunas.

No entanto, esses pontos não são suficientes para criar uma representação bidimensional precisa do ambiente, pois, podem apresentar ruídos, erros de medição e variação da posição do sensor em relação ao ambiente. Para superar essas limitações, as leituras devem ser processadas por meio de filtros e algoritmos para a construção de um mapa de dados e, então, ser possível a localização do drone em tempo real.

Para essa finalidade, foi utilizado um pacote do ROS2 (*Robot Operating System Version 2*) chamado SLAM (*Simultaneous Localization and Mapping*). Esse pacote permite disponibilizar as leituras de ângulo e distância em um ambiente fechado para a criação de um mapa local, como o Laboratório de Drones do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina - Florianópolis. Aplicando seu algoritmo e uma série de transformações geométricas, foi possível gerar uma imagem de 8 bits em escala de cinza representando um mapa local. Este mapa está ilustrado na Figura 8.

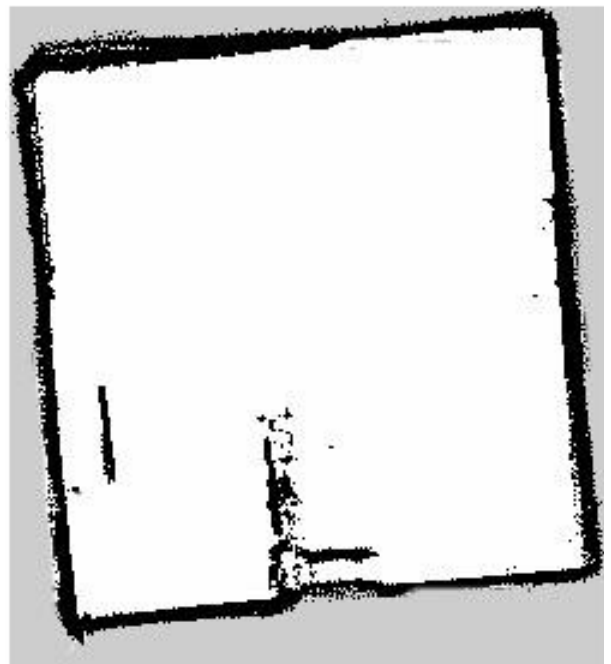


Figura 8. Mapa do Laboratório de drones IFSC gerado pelo SLAM.



Cada valor de pixel no mapa gerado apresenta informações específicas sobre o ambiente. Os pixels em cinza indicam áreas desconhecidas, onde o algoritmo SLAM não conseguiu identificar dados. Os pixels brancos representam espaços livres, sem obstáculos, enquanto os pixels pretos correspondem a objetos sólidos identificados, como paredes ou estantes.

Além de fornecer uma representação visual útil para a análise do operador, os dados de cor, distância e posição de cada pixel são medidas reais e em escala, essenciais para diversas aplicações. Esses dados são utilizados por algoritmos responsáveis pelo planejamento de rotas e controle dos atuadores de robôs autônomos, entre outras funções.

O lidar rotativo será o principal sensor de odometria utilizado pela equipe no drone. No entanto, outros sensores também auxiliarão na operação de deslocamento e coordenação do veículo, como o sensor HereFlow e os sensores inerciais.

#### IV. SISTEMA DE DETECÇÃO E POUSO NAS BASES

Nesta seção, é apresentada a estratégia implementada para a detecção das bases e o processo de pouso preciso, utilizados na Fase 1: Localização e Mapeamento das Bases. Esta fase da competição envolve a detecção e distinção de bases de pouso. Para alcançar esse objetivo, foi desenvolvido um sistema de visão computacional e pouso de precisão.

O sistema implementado utiliza a biblioteca OpenCV (*Open Source Computer Vision Library*), uma biblioteca de código aberto focada em visão computacional e processamento de imagens [7]. O OpenCV oferece uma ampla gama de funcionalidades para a manipulação, análise e processamento de imagens em tempo real. Adicionalmente, foi empregada a biblioteca TensorFlow, desenvolvida pelo Google, especialmente projetada para tarefas de aprendizado de máquina e deep learning [8]. O TensorFlow permite a criação e treinamento de redes neurais para resolver problemas complexos, como a detecção de objetos.

O objetivo do sistema é permitir que o drone reconheça as bases de pouso em imagens obtidas por uma câmera embarcada, utilizando o OpenCV para captura e processamento de imagens em tempo real, e o modelo de rede neural treinado pela equipe para realizar inferências e detecções das diferentes bases de pouso. O processo geral para a implementação deste sistema envolve as seguintes etapas:

##### 1. Coleta de Dados

Primeiramente, foi necessário coletar um conjunto significativo de imagens contendo diferentes ângulos, condições de iluminação e variações dos objetos a serem reconhecidos. Quanto mais diversificado e representativo for o conjunto de dados, melhor será o desempenho esperado do sistema de reconhecimento.

##### 2. Pré-processamento das Imagens

As imagens coletadas precisaram ser pré-processadas antes de serem utilizadas no treinamento da rede neural. Esse pré-processamento incluiu o redimensionamento das imagens para um tamanho padrão, a normalização dos valores dos pixels e a aplicação de técnicas de aumento de dados (*data augmentation*)

para aumentar a quantidade de amostras disponíveis. O conjunto de dados original foi expandido em 17 vezes na sua totalidade.

##### 3. Treinamento da Rede Neural

Com as imagens pré-processadas e devidamente anotadas, foi possível treinar uma rede neural usando o TensorFlow. Essa rede foi projetada para detectar a classe de interesse nas imagens recebidas pela câmera. A utilização de uma arquitetura de rede neural convolucional (CNN) é comum para esse tipo de tarefa, dada sua eficiência em tarefas de detecção de objetos em imagens.

##### 4. Implementação no Drone

Após o treinamento, a rede neural foi avaliada e validada utilizando diferentes métricas, como a matriz de confusão, para assegurar o sucesso do processo de treinamento. Uma vez validada, a rede neural foi implementada no drone para realizar inferência em tempo real nas imagens capturadas pela câmera embarcada.

Durante a operação do drone, a câmera captura continuamente imagens do ambiente. Essas imagens são processadas e enviadas para a rede neural, que realiza a detecção em tempo real. Esse processo permite que o drone reconheça e responda a objetos e condições do ambiente de maneira eficiente e precisa.

##### 5. Decisão de Pouso

Com base nas previsões da rede neural, o drone pode tomar decisões em relação ao pouso. Se a base de pouso desejada for detectada com um valor de confiança pré estabelecido, o drone pode prosseguir com o procedimento de pouso. Caso contrário, ele pode permanecer no estado anterior e procurar por outra base ou adotar uma estratégia para aguardar por uma melhor detecção.

##### 6. Pouso de Precisão

Uma vez que o drone tenha iniciado o processo de pouso, um algoritmo para o pouso de precisão utiliza a posição da base de pouso detectada em relação ao frame total da imagem para corrigir a posição XY do drone. Além desta correção, o algoritmo também considera o posicionamento do drone no eixo Z, ajustando cuidadosamente sua altura para garantir que o pouso seja suave, evitando assim danos ao protótipo.

## AGRADECIMENTOS

Gostaríamos de expressar nossa gratidão à FEESC e à FAPESC por recursos utilizados neste projeto. Agradecemos também ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina (IFSC - Florianópolis) pela infraestrutura disponibilizada e pela oportunidade de desenvolver este projeto. O ambiente de pesquisa e os recursos fornecidos pelo IFSC Florianópolis foram essenciais para a realização de nossos objetivos. Reconhecemos a dedicação e o apoio de todos os envolvidos, que contribuíram significativamente para o desenvolvimento e a conclusão deste trabalho.

## REFERÊNCIAS

- [1] Ardupilot Hex HereFlow Optical Flow Sensor, <https://ardupilot.org/copter/docs/common-hereflow.html>, último acesso em 23/07/2024
- [2] Ardupilot Rangefinders (landing page), <https://ardupilot.org/copter/docs/common-rangefinder-landingpage.html>, último acesso em 23/07/2024
- [3] J. M. A. Chaves, J. P. A. Duarte, M. E. Gonçalves, et al.: Delta High Team Description Paper. CBR, 2023
- [4] MAVLink Developer Guide, <https://mavlink.io/en/>, último acesso em 23/07/2024
- [5] GitHub Pymavlink, <https://github.com/ArduPilot/pymavlink>, último acesso em 23/07/2024
- [6] GitHub Delta High Team, <https://github.com/Delta-High-Team>, último acesso em 23/07/2024
- [7] OpenCV, <https://opencv.org/>, último acesso em 23/07/2024
- [8] Introdução ao TensorFlow, <https://www.tensorflow.org/?hl=pt-br>, último acesso em 23/07/2024