



**INSTITUTO
FEDERAL**
Brasília

Instituto Federal de Brasília
Campus Taguatinga
Tecnologia em Automação Industrial

ABIMAE L IBAINO DE ALMEIDA

**DESENVOLVIMENTO DE UM PROTÓTIPO:
DRONE PARA RECONHECIMENTO DE PESSOAS**

Brasília
2025

ABIMAE L IBAINO DE ALMEIDA

**DESENVOLVIMENTO DE UM PROTÓTIPO:
DRONE COM IDENTIFICAÇÃO DE PESSOAS**

Trabalho de Conclusão de Curso apresentado ao curso de Tecnologia em Automação Industrial do Instituto Federal de Brasília, Campus Taguatinga como parte da exigência para obtenção do título de tecnólogo.

Orientador(a): Doutor Marcus Vinicius Costa Sá

Brasília
2025

Ficha de identificação da obra elaborada pelo bibliotecário
Marcelo José Rodrigues da Conceição (CRB1-2323)

Almeida, Abimael Libaino de

A447d Desenvolvimento de um protótipo: drone capaz de reconhecimento de pessoas /
Abimael Libaino de Almeida. Brasília-DF, 2025.

60 f. : il.

Trabalho de Conclusão de Curso (Tecnologia de Automação Industrial) Instituto
Federal de Educação, Ciência e Tecnologia de Brasília, Campus Taguatinga, 2025.

Orientador: Marcus Vinicius Costa Sá.

Inclui referências.

1. Automação industrial. 2. Drone. 3. Inteligência artificial. 4. Sistemas embarcados
(Computadores). 5. Software - Desenvolvimento. I. Título. II. Sá, Marcus Vinicius Costa.
III. Instituto Federal de Educação, Ciência e Tecnologia de Brasília.

CDU 681.5:004

Dedico esse trabalho à minha família e amigos que foram apoio e incentivo na realização do mesmo.

AGRADECIMENTOS

Agradeço a oportunidade dada por meus colegas e professores de entrar em projetos e atividades que possibilitaram a aquisição de conhecimentos. Entre esses, o meu orientador e professor Marcos, que me oportunizou participar dos seus projetos de extensão. Também aos meus amigos e colegas Johan, David, Tales e Samuel, que me auxiliaram e me acolheram como contemporâneo e amigo, e meus outros professores que sem suas aulas e transmissão de conhecimento, não conseguiria chegar aqui. Finalmente, mais importante ainda, agradeço a minha família que me deu condições de buscar por minhas aptidões com calma e apreço as minhas opiniões. Seu auxílio e ajuda foram fundamentais para eu me tornar a pessoa que sou hoje.

A persistência é o caminho do êxito.

— **Charles Chaplin.**

RESUMO

Este trabalho tem como objetivo geral elaborar um protótipo para ser usado como plataforma de teste de modelos de identificação de objetos em drones. Para isso, foram estabelecidos objetivos específicos como a implementação de um sistema de processamento de dados capaz de identificar pessoas usando um drone com computador embarcado e o aprimoramento do entendimento de como elaborar um drone para auxílio em missões de resgate. A metodologia envolveu a pesquisa e aquisição das peças para o drone, montagem do protótipo, calibração e ajustes necessários, integração do sistema a um computador embarcado e testes de voo em diferentes condições. Os resultados mostraram que o protótipo é eficaz na identificação de pessoas e pode ser uma ferramenta valiosa em operações de busca e resgate. Conclui-se que o uso de drones com inteligência artificial embarcada pode aumentar a eficiência e precisão dessas operações.

Palavras-chave: drones; inteligência artificial; computador embarcado.

ABSTRACT

This work aims to develop a prototype to be used as a testing platform for object identification models in drones. To achieve this, specific objectives were established, such as the implementation of a data processing system capable of identifying people using a drone with an onboard computer and the improvement of the understanding of how to develop a drone to assist in rescue missions. The methodology involved researching and acquiring parts for the drone, assembling the prototype, calibration and necessary adjustments, integrating the system with an onboard computer, and conducting flight tests under different conditions. The results showed that the prototype is effective in identifying people and can be a valuable tool in search and rescue operations. It is concluded that the use of drones with embedded artificial intelligence can increase the efficiency and accuracy of these operations.

Keywords: drones; artificial intelligence; companion computer.

LISTA DE ILUSTRAÇÕES

Figura 1 - Modelo de uma renderização de um hexacoptero	16
Figura 2 - Modelo do drone usado como base para o trabalho	16
Figura 3 - Site da comunidade <i>Ardupilot</i>	18
Figura 4 - <i>Mission Planner</i>	18
Figura 5 - Placa controlador a <i>Pixhawk 6C</i>	19
Figura 6 - <i>Raspberry pi 5B</i> com Kit IA	20
Figura 7 - Exemplo de uso para o protocolo RTSP	21
Figura 8 - Ampliação de um pixel em um monitor	23
Figura 9 - Pixel representado usando o protocolo RGB	23
Figura 10 - Formato da mensagem MAVLink	24
Figura 11 - Rede CNN representada como gráfico	25
Figura 12 - Homem sendo resgatado em Canoas-RS	27
Figura 13 - Helicóptero dos Bombeiros sobrevoando Brumadinho depois do rompimento da barragem	27
Figura 14 - Exemplo da medição da massa.	30
Figura 15 - Kit Frame X500 V2 em sua caixa	32
Figura 16 - Braços do drone montados	33
Figura 17 - Parte inferior do frame montada.....	34
Figura 18 - Drone básico montado em meio de teste de sentido dos motores.....	34
Figura 19 - Imagem do protótipo	35
Figura 20 - Suporte para a câmera Siyi A2 Mini.....	36
Figura 21 - Pinout dos conectores e especificações da câmera Siyi A2 Mini.....	37
Figura 22 - Pinout do conector RJ45	38
Figura 23 - Página de seleção do tipo de instalação do firmware	41
Figura 24 - Página de seleção do tipo de frame	41
Figura 25 - Página de calibração do compass	42
Figura 26 - Página de calibração do acelerômetro	42
Figura 27 - Página de teste dos motores	43
Figura 28 - Página de calibração parâmetros iniciais.....	43
Figura 29 - Parte do código para controle do gimbal da SIYI A2 Mini	46
Figura 30 - Parte do código para registrar a localização e quantidade de pessoas na foto em arquivo JSON	46
Figura 31 - Conexão UART entre Raspberry PI e Pixhawk.....	47
Figura 32 - Quantidade de modelos e modificações no dataset realizados	51
Figura 33 - Foto de uma das filmagens com o YOLOv8s em baixa luminosidade	52

Figura 34 - Foto de uma das filmagens com o YOLOv8s em alta luminosidade52

LISTA DE TABELAS

Tabela 1 - Resultados da medição de massa	31
Tabela 2 - Relação da conexão necessária para a adaptação	39

LISTA DE ABREVIATURAS E SIGLAS

CNN	<i>Convolutional Neural Network</i>
ESC	<i>Electronic Speed Controller</i>
FCU	<i>Flight Control Unit</i>
GCS	<i>Ground Control Unit</i>
GNSS	<i>Global Navigation Satellite System</i>
GND	<i>Ground</i>
GPS	<i>Global Positioning System</i>
IA	Inteligência Artificial
IMU	<i>Inercial Measurement Unit</i>
NPU	<i>Neural Processing Unit</i>
PID	Proporcional-Integral-Derivativo
RTSP	<i>Real-Time Streaming Protocol</i>
SDK	<i>Software Development Kit</i>
UAV	<i>Unmanned Aerial Vehicle</i>
VANT	Veiculos Aéreos Não Tripulados
VCC	<i>Voltage Common Collector</i>
YOLO	<i>You Only Look Once</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivo Geral	14
1.2	Objetivos Específicos	14
1.3	Justificativa.....	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	VANT – Veículo Aéreo Não Tripulado.....	15
2.2	Plataforma Ardupilot.....	17
2.3	Pixhawk	19
2.4	Raspberry Pi	20
2.5	Protocolo RTSP	21
2.6	Visão Computacional.....	22
2.7	Protocolo MAVLink	24
2.8	Redes Neurais.....	25
2.9	Utilização de Drones em Situações de Salvamento	26
2.10	Revisão Bibliográfica	28
3	METODOLOGIA	30
3.1	Testes em Bancada	30
3.2	Montagem	32
3.2.1	<i>Montagem Básica</i>	<i>32</i>
3.2.2	<i>Modelagem dos Suportes.....</i>	<i>36</i>
3.2.3	<i>Adaptação da Câmera SIYI A2 Mini</i>	<i>37</i>
3.3	Configuração	40
3.3.1	<i>Configurações Básicas no Mission Planner</i>	<i>40</i>
3.3.2	<i>Escolha da Placa a Ser Embarcada</i>	<i>44</i>
3.3.3	<i>Calibração e Ajuste do PID.....</i>	<i>45</i>
3.3.4	<i>Modificação do código para uso no drone.....</i>	<i>45</i>
3.3.5	<i>Comunicação Entre Raspberry Pi/Jetson Nano e Pixhawk</i>	<i>47</i>
3.3.6	<i>Retreinamento de modelos YOLO</i>	<i>49</i>
4	RESULTADOS E DISCUSSÕES	50
5	CONCLUSÃO	53
	REFERÊNCIAS.....	54

1 INTRODUÇÃO

Os Veículos Aéreos Não Tripulados (VANTs), comumente conhecidos como drones, têm se destacado como uma das inovações tecnológicas mais versáteis e promissoras das últimas décadas. Controláveis em três eixos sem a necessidade de pilotos a bordo, essas aeronaves abrangem desde drones multirotores, que utilizam variações de torque dos motores para controlar movimentos, até modelos de asa fixa, empregados em operações de maior alcance e duração. Sua popularidade crescente deve-se, em grande parte, à diversidade de tamanhos, configurações e aplicações que oferecem, abrangendo desde corridas recreativas até transporte de cargas, inspeções industriais e operações de busca e resgate.

Historicamente, como descrito em *“A Brief History of Early Unmanned Aircraft”* por John Keane e Stephen Carr, as pesquisas para o desenvolvimento de drones remontam à Primeira Guerra Mundial, quando os primeiros esforços buscavam criar aeronaves não tripuladas, como o *Kettering Bug* e torpedos aéreos, capazes de realizar missões perigosas sem colocar pilotos em risco. Durante os períodos seguintes, incluindo a Segunda Guerra Mundial e a Guerra Fria, os UAVs (veículos aéreos não tripulados) evoluíram significativamente, passando de simples armas experimentais e drones-alvo para plataformas mais sofisticadas de inteligência, vigilância e reconhecimento. Apesar de sua aplicação predominante em contextos de conflitos bélicos devido à sua capacidade de reduzir riscos humanos em missões de alto perigo, essas aeronaves demonstraram um vasto potencial para usos não militares. Tecnologias inicialmente projetadas para combates, como controle remoto e sensores avançados, foram adaptadas ao longo do tempo, abrindo caminho para a aplicação de drones em áreas como monitoramento ambiental, agricultura de precisão, inspeções industriais e até mesmo logística. Este desenvolvimento destaca a versatilidade dos UAVs, que transcendem suas origens militares para atender a diversas demandas modernas.

Em 2021, o Corpo de Bombeiros do Distrito Federal (CBMDF) anunciou aquisição de drones para serem utilizados durante as operações de resgate. Tais aeronaves equipadas para missões noturnas, incêndios florestais, grandes eventos, entre outros. Tal publicação informa que são mais eficazes pois trazem imagens mais precisas para as diversas operações.

1.1 Objetivo Geral

Este trabalho teve como objetivo geral elaborar um protótipo para ser usado como plataforma de teste de modelos de identificação de objetos em drones.

1.2 Objetivos Específicos

Para aprimorar o desenvolvimento deste tema, foram estabelecidos os seguintes objetivos específicos para este trabalho:

1. Especificar e montar um drone capaz de se movimentar enquanto o computador embarcado funcionasse.
2. Implementar um sistema de processamento de dados capaz de identificar pessoas usando um drone com computador embarcado.

1.3 Justificativa

No contexto de busca e resgate, os drones de asa fixa já demonstram grande potencial ao cobrir grandes áreas e registrar vídeos que auxiliam na identificação de pessoas que estejam em situações emergenciais. Exemplos desse uso emergem em contextos reais, como o recente auxílio de drones nas enchentes no Rio Grande do Sul no Brasil, e em projetos experimentais, como os testes conduzidos pela UNICEF no Cazaquistão (2019). Nessas situações, drones superam adversidades climáticas e geográficas, contribuindo significativamente para localizar vítimas em áreas de difícil acesso. Tais avanços dependem de configurações tecnológicas robustas, que incluem câmeras de alta definição, *gimbals* estabilizadores e sistemas embarcados capazes de processar dados complexos em tempo real.

No entanto, o processo tradicional, que exige o envio de imagens para uma estação terrestre onde ocorre a análise visual, pode ser lento e dependente da intervenção humana. Assim, surge a possibilidade de aprimorar esse processo com o uso de inteligência artificial (IA) embarcada. Se equipados com sistemas de aprendizado de máquina, drones podem identificar automaticamente pessoas em imagens capturadas e enviar somente os trechos relevantes para verificação. Isso aumentaria a eficiência e a precisão das operações. Por isso se justifica o presente trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 VANT – Veículo Aéreo Não Tripulado

As aeronaves que não possuem um piloto embarcado são geralmente classificadas como Veículo Aéreo Não Tripulado (VANT). Segundo Quan Quan em *Introduction to Multicopter Design and Control*, VANTs são mais do que simples aeromodelos. Devido ao fato de serem capazes de realizar atividades além das recreativas, com o auxílio de computadores embarcados em veículos semiautônomos ou autônomos, o que os tornam mais inteligentes.

O avanço dos drones multicópteros, ou drones de asa rotativa, teve seu início em torno de 1950 com o desenvolvimento computacional. Por serem aerodinamicamente instáveis, os drones multicópteros precisam de sensores para reagir às variáveis ambientais, principalmente as unidades de medida inercial (*IMU*) e os sistemas globais de navegação por satélite (*GNSS*). Esses recursos proporcionam alta precisão, capacidade de pairar no ar e melhor manobrabilidade, superando os aspectos negativos.

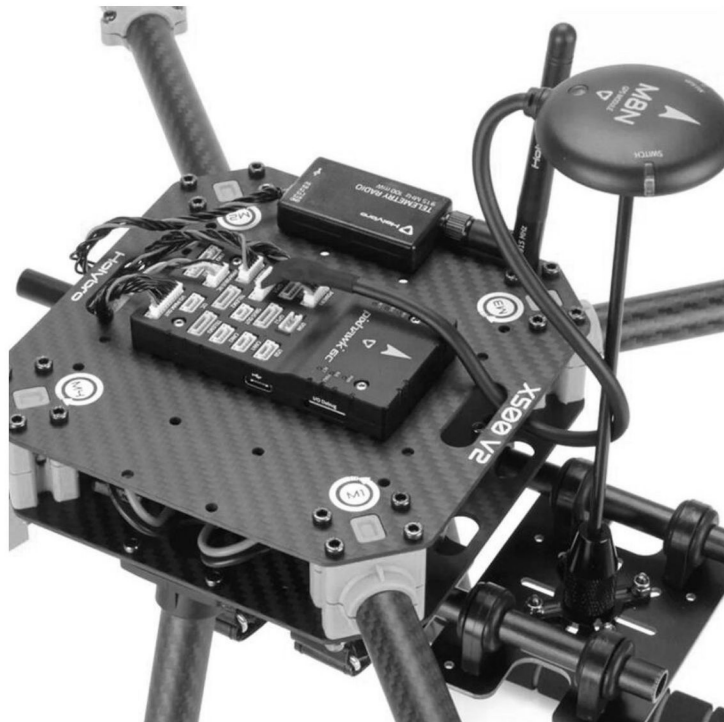
A redundância de controle e sensores nesses veículos é crucial para evitar falhas. Entre os tipos de multicópteros, o quadricóptero (multicopter de quatro rotores) destaca-se pela facilidade de montagem, manutenção e preço reduzido. No entanto, a falha de um motor pode comprometer ou inutilizar a aeronave. Um exemplo dessas aeronaves é mostrado na Figura 1, em específico, um hexacóptero. A Figura 2 mostra um modelo de quadricóptero que foi utilizado como base para esse projeto.

Figura 1 - Modelo de uma renderização de um hexacoptero



Fonte: Autores, 2022.

Figura 2 - Modelo do drone usado como base para o trabalho



Fonte: Adaptado do Reichelt^[1] (2024)

2.2 Plataforma Ardupilot

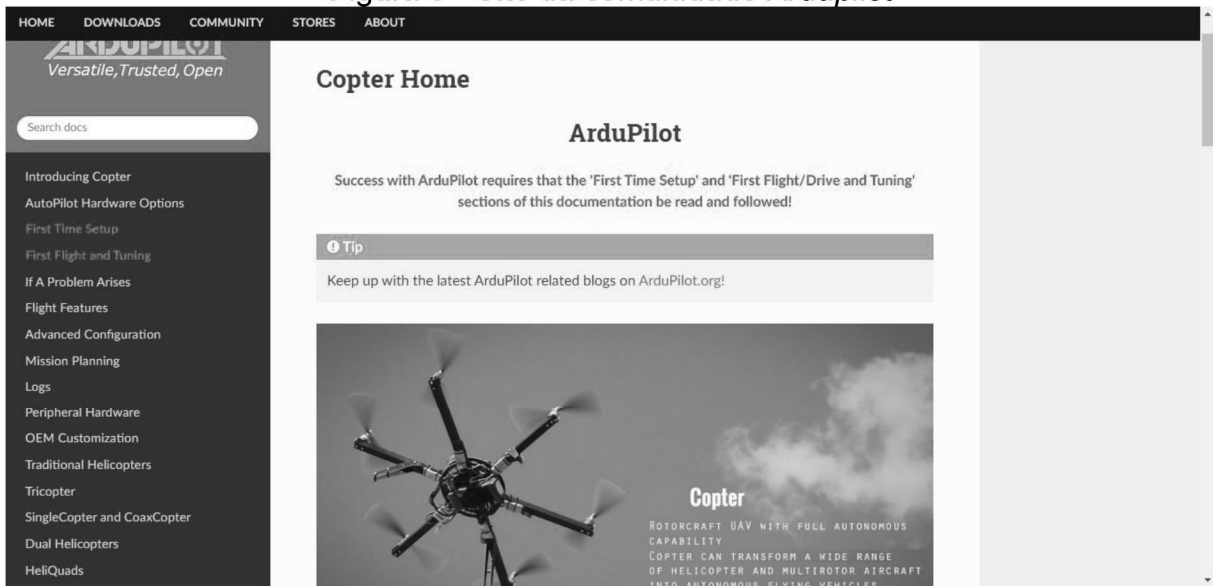
Ardupilot é um projeto *opensource* para elaboração de drones inaugurado por Chris Anderson e Jordi Munoz em 2009. Essa plataforma foi pensada para suportar diferentes tipos de modos de veículos e dispositivos passíveis de automação, são eles: *rover* (veículo terrestre), *copter* (asa rotativa), *plane* (asa fixa), *sub* (submersos) e *Antenna Tracker* (antena seguidora).

A plataforma é uma referência de *autopilots* hoje composta por uma das maiores redes de desenvolvedores de diversas áreas da engenharia, incluindo tanto *hardware* quanto *software*, tendo um site com documentação que foi alimentada por mais de 10 anos, mostrado na figura 3.

Embora o projeto possuísse licença GPLv3, que é gratuita, exige contribuição obrigatória em caso de modificação, impedindo que empresas e pessoas se apropriem do projeto ao implementar algum recurso sem dar algo em troca à comunidade.

Para auxiliar na configuração dos drones, a plataforma conta com o auxílio de algumas *GCS* (*Ground Control Station*), que são estações de controle terrestres com funcionalidades que facilitam a calibração e interação de dispositivos, sensores, como também parâmetros e planejamento de missões autônomas. O *Mission Planner* é o mais completo e popular, como mostrado na Figura 4. Por esses motivos, foram selecionadas as plataformas e a estação de solo.

Figura 3 - Site da comunidade Ardupilot



Fonte: Ardupilot, 2021.

Figura 4 - Mission Planner

WP	Radius	Loiter Radius	Default Alt	Absolute Alt	Verify Height	Lat	Long	Alt	Delete	Up	Down	Grad %	Dist	AZ
1	WAYPOINT	0	0	0	0	-35.0407928	117.8277898	100	X	⬆️	⬆️	95.7	104.5	1
2	WAYPOINT	0	0	0	0	-35.0406786	117.8260410	100	X	⬆️	⬆️	0.0	159.7	275
3	WAYPOINT	0	0	0	0	-35.0417239	117.8251612	100	X	⬆️	⬆️	0.0	141.2	215
4	WAYPOINT	0	0	0	0	-35.0428395	117.8259873	100	X	⬆️	⬆️	0.0	145.1	149
5	WAYPOINT	0	0	0	0	-35.0427165	117.8274572	100	X	⬆️	⬆️	0.0	134.5	84

Fonte: Ardupilot, 2021.

2.3 Pixhawk

Pixhawk é um padrão *opensource* de placas controladoras específicas para drones. Esses padrões providenciam especificações de *hardware*, seja na parte da mecânica quanto da elétrica, e guias a serem seguidos na confecção ou do mesmo.

Esse padrão é desenvolvido pela equipe *Pixhawk Special Interest Group* que é coordenada pela *Dronecode Foundation*.

A base é um projeto de pesquisa criado por Lorenz Meier que na época era um aluno de mestrado em ETH Zurich. Inicialmente, na sua criação foram usados *frameworks* e *drivers opensource* o que possibilitou a distribuição do *software* de forma gratuita a todos, tanto que algumas ferramentas criadas nesse projeto são usadas até hoje no *Ardupilot* como *MAVLink* (Auterion, 2020).

Hoje em dia, existem padrões disponíveis para a plataforma *Pixhawk* e para as placas de controle de potência e placas controladoras. Na Figura 5 abaixo, temos um exemplo de placa controladora *Pixhawk*, feita pela empresa *Holybro*.

O que torna os novos modelos especiais são as redundâncias de sensoriamento, transformando os possíveis grandes erros de cálculo que tornam o drone instável em problemas cada vez menores a depender do grau de redundância. Por esse motivo, foi escolhido o modelo *Pixhawk 6C* da *Holybro*.

Figura 5 - Placa controladora *Pixhawk 6C*



Fonte: Holybro, 2022.

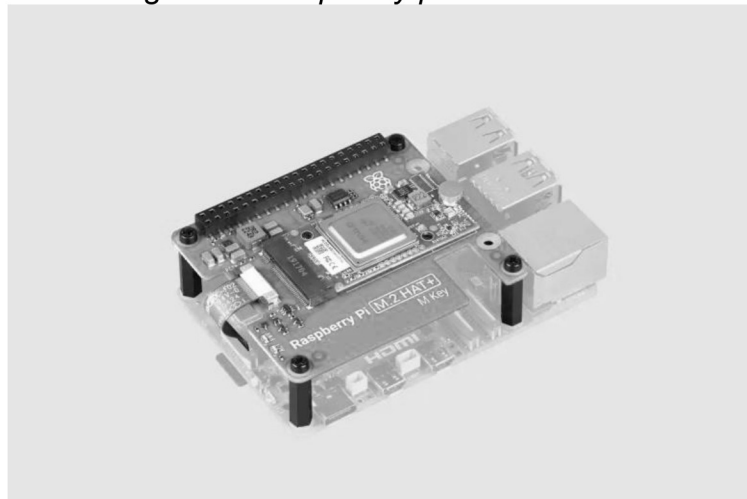
2.4 Raspberry Pi

O *Raspberry Pi* é um microcomputador que já vem com todas as partes necessárias integradas numa só placa de fenolite. Ele foi desenvolvido pela Fundação *Raspberry Pi* no Reino Unido. A ideia de criar tal dispositivo foi impulsionada pelo aparente declínio do nível de conhecimento dos alunos na universidade de Cambridge percebido por um grupo de estudantes da área de computação.

Inicialmente, foi criado para ser utilizado como auxílio para o ensino de conhecimentos de computação básica nas escolas de países em desenvolvimento, mas a utilização cresceu de tal maneira que hoje em dia são utilizados até em meios industriais. Isso se dá pelo fato que é supostamente uma solução de baixo custo para projetos pessoais, ser mais documentada e aplicada para diversas atividades e por ter uma enorme comunidade de desenvolvedores. Por esses motivos que para o projeto, utilizou-se uma *Raspberry Pi 5 model B* (Figura 6).

Durante a execução do projeto, deu-se o lançamento do kit IA da própria *Raspberry Pi Foundation*. Ela consegue com ajuda de uma NPU (*Neural Processing Unit*) fazer reconhecimento de objetos em uma taxa de entre 24 e 30 quadros por segundo usando modelos YOLOv6n e YOLOv8s padrões convertidos para hef.

Figura 6 - *Raspberry pi 5B* com Kit IA

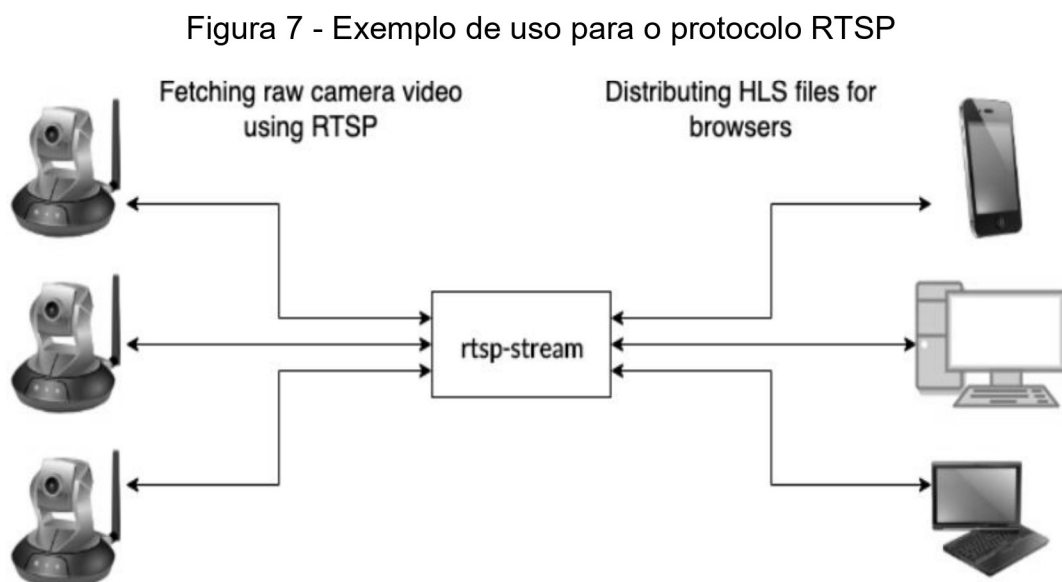


Fonte: *Raspberry Pi LTD* (2024).

2.5 Protocolo RTSP

O Protocolo de Streaming em Tempo Real (RTSP) é um padrão de comunicação de rede que permite o controle e a entrega de fluxos de mídia em tempo real. Desenvolvido pela IETF em 1998, o RTSP foi projetado para estabelecer e controlar sessões de mídia entre pontos de extremidade. Ele é comumente utilizado em sistemas de câmeras de segurança IP, nos quais facilita a transmissão e o controle de vídeo e áudio ao vivo, ou em câmeras da marca SIYI. Ao contrário de outros protocolos que se concentram na entrega de conteúdo, o RTSP se destaca por permitir o controle interativo do fluxo de mídia. Possui as funções de pausar, retroceder ou avançar o conteúdo transmitido, o que é essencial para aplicações como vigilância por vídeo e teleconferência.

O RTSP opera sobre os protocolos de transporte TCP ou UDP e geralmente usa a porta 554. Uma das principais vantagens do RTSP é sua capacidade de suportar a transmissão de múltiplos fluxos de mídia simultaneamente, o que é benéfico para apresentações complexas que incluem diferentes tipos de mídia. Além disso, o RTSP é altamente compatível com uma variedade de dispositivos e players de mídia, tornando-o uma escolha versátil para muitas aplicações de streaming.



Fonte: Tonmind (2024).

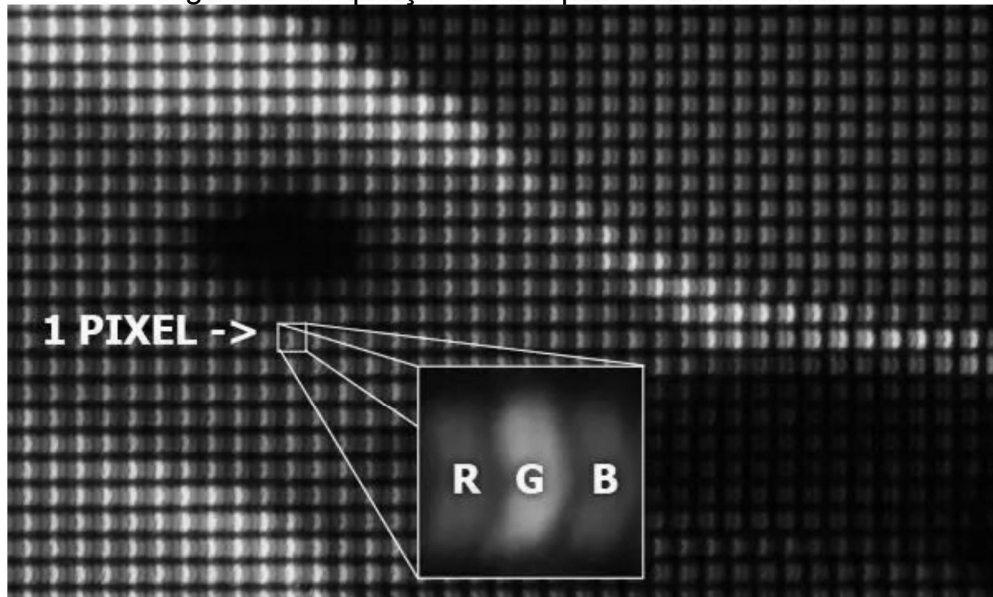
2.6 Visão Computacional

A Visão Computacional, ou Visão Robótica, é ponto chave para o desenvolvimento da robótica móvel. Suas decisões são tomadas baseadas no que ele pode "enxergar", com uma semelhança da percepção humana de ver o ambiente ao seu redor, o que se torna uma ferramenta extremamente poderosa. O sensor de uma câmera, que atua parecido com os olhos humanos, capta uma imagem e a converte para informações discretizadas, ou seja, em números, de forma que o computador entenda. A imagem em um monitor é formada por uma matriz de pixels, sendo cada pixel formado por três cores: vermelho, verde e azul, ou a sigla *RGB* em inglês, ilustrado na Figura 8. Com esse pixel podemos formar qualquer cor visível ao olho humano manipulando cada cor de 0 a 255, como mostrado na figura 9.

A OpenCV é uma biblioteca com suporte para Python e C++, *opensource* e gratuita, voltada para aplicações de visão computacional. Ela possui mais de 2500 algoritmos otimizados variados, cuja aplicação pode ser a detecção de objetos, reconhecimento facial, rastreamento de objetos, extração de modelos de objetos, dentre muitas outras serventias.

Dessa forma, a biblioteca OpenCV está entre as que mais contém recursos com objetivo de além de detectar objetos, os reconhecer e saber diferenciar de outros, por meio de *machine learning* e *deep learning*. Um dos algoritmos mais otimizados para isso é o YOLO, também usado no projeto, que faz análise com problema de regressão e fornece as probabilidades de classe das imagens detectadas, tudo isso em tempo real. A comunidade OpenCV, para visão computacional, é a mais ativa atualmente, tornando-a mais adequada para o projeto.

Figura 8 - Ampliação de um pixel em um monitor



Fonte: MOTA, 2018.

Figura 9 - Pixel representado usando o protocolo RGB

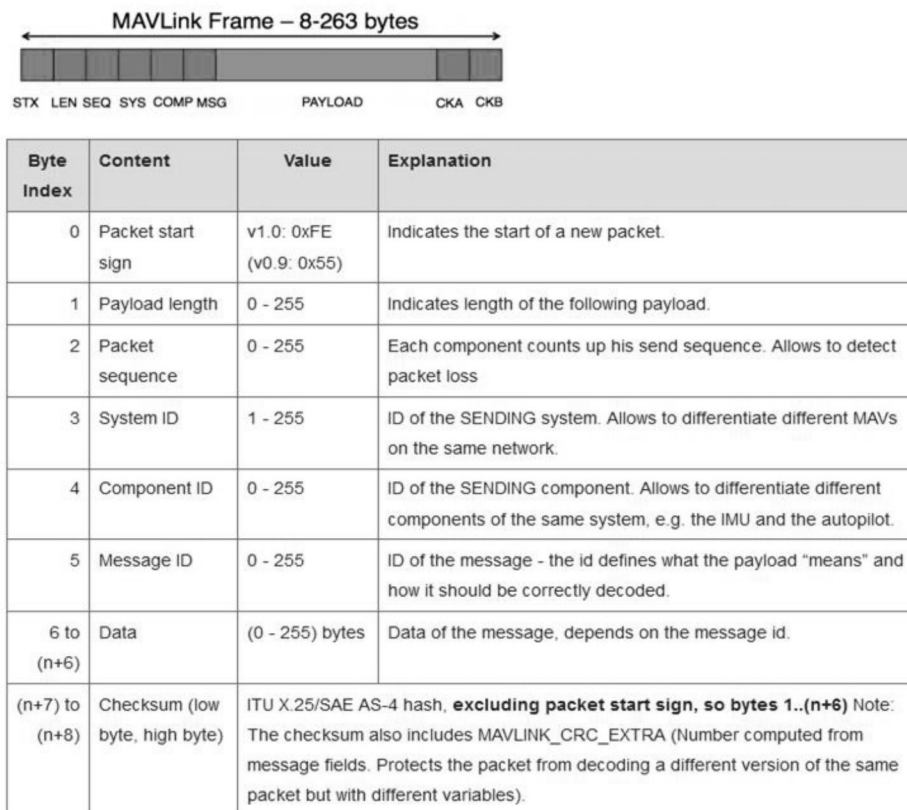
R:	255	0	0	0	255
G:	0	255	0	0	255
B:	0	0	255	0	255

Fonte: MOTA, 2018.

2.7 Protocolo MAVLink

Um protocolo serial comumente usado para comunicação entre veículos, estação de solo (GCS) e componentes integrados é o MAVLink. Sua função é ser uma comunicação leve, pela necessidade de ser implementado em dispositivos com restrição de recursos, e segura, para que sejam evitados interferências e pacotes de dados incompletos. As mensagens MAVLink podem ser enviadas por praticamente qualquer conexão serial. Outro recurso importante do MAVLink é seu *design multicast*, que permite a transmissão de dados entre múltiplos destinatários diferentes. A composição do formato da mensagem é ilustrada na figura 10.

Figura 10 - Formato da mensagem MAVLink



Fonte: ArduPilot, 2021.

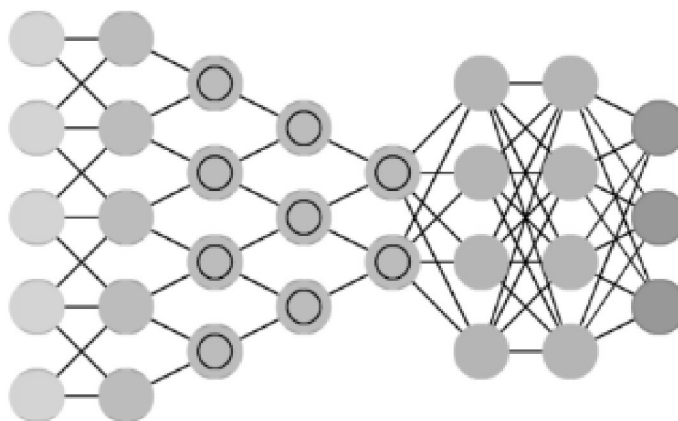
2.8 Redes Neurais

Uma rede neural é uma solução computacional que utiliza cálculos e funções matemáticas para imitar o funcionamento do cérebro humano em tarefas como classificação, previsão e reconhecimento, entre outras. Seu funcionamento é baseado em neurônios artificiais, ou “nós”, que transmitem dados entre si. Por meio de pesos nas ligações entre os neurônios, funções de perda e parâmetros predefinidos, o programa ajusta esses pesos para alcançar o resultado desejado.

A estrutura de uma rede neural pode ser simplificada como composta por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada nó está conectado a outros e possui seu próprio peso e limiar. Se a saída de um nó individual ultrapassar o valor de limiar especificado, esse nó é ativado e envia dados para a próxima camada da rede. Caso contrário, nenhum dado é transmitido para a camada seguinte.

O modelo de rede neural utilizado é o YOLO (*You Only Look Once*), baseado na arquitetura *Convolutional Neural Network* (CNN), ou Rede Neural Convolutiva. Sua diferenciação no funcionamento ocorre nas camadas de convolução, onde a informação passa por diversos filtros que destacam padrões locais regulares, enquanto reduzem a dimensão dos dados originais. Os resultados de vários filtros são resumidos por operações de *pooling*. Nas camadas mais profundas das convoluções, espera-se que os dados, em um espaço dimensional reduzido, contenham informações suficientes sobre esses padrões locais para atribuir um valor semântico aos dados originais. Esses dados são então processados por uma estrutura mais simples para a tarefa de classificação.

Figura 11 - Rede CNN representada como gráfico



2.9 Utilização de Drones em Situações de Salvamento

Hoje em dia, tornou-se comum o uso de drones de asa fixa equipados com vários tipos de sensores para fazer um voo rasante à procura de pessoas. Isso se deu no episódio das enchentes do Rio Grande do Sul em 2024. Tais equipamentos são adequados pois dispensam a necessidade de pilotos de helicóptero para fazerem esses voos no reconhecimento. Isso os libera para se preocupar mais com o resgate de pessoas nessa situação. Entretanto, ainda não existe no mercado uma variedade desse tipo de drones que são melhores para serem usados em áreas abertas. Um possível avanço seria usar multicópteros para fazer um reconhecimento mais apurado após o reconhecimento preliminar com o drone de asa fixa. Assim, poderiam ser usados em áreas menos acessíveis e diminuir o perigo de socorristas terem que entrar em lugares de difícil acesso e arriscados no intuito de encontrar pessoas e resgatá-las. Assim sendo, a entrada em tais áreas se daria apenas para o resgate, o que diminuiria o tempo e riscos da operação.

Um exemplo de tais situações seria em florestas onde pessoas se perdem. Estudos, como por exemplo, *The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations* por Karaka em sua conclusão afirma o seguinte: “uma área mais ampla pode ser pesquisada mais rapidamente por drone usando DST em comparação com a técnica clássica, e a vítima pode ser localizada mais rapidamente e alcançada mais cedo (...)” demonstrando a eficiência dessas tecnologias.

O projeto realizado pelo autor visava a implementação de um sistema de visão computacional e inteligência artificial para reconhecimento de pessoas que fosse mais confiável e que os drones pudessem ser usados em áreas mais longas e sem conexão à internet.

Figura 12 - Homem sendo resgatado em Canoas-RS



Fonte: BBC, 2024.

Figura 13 - Helicóptero dos Bombeiros sobrevoando Brumadinho depois do rompimento da barragem



Fonte: RIBEIRO et al, 2019.

2.10 Revisão Bibliográfica

Vale ressaltar a existência de trabalhos sobre a utilização de drones com capacidade de detecção de objetos. Dentre esses destacam-se primeiramente "*Convolutional Neural Network-Based Real-Time Object Detection and Tracking for Parrot AR Drone 2*" por Rohan Ali et al, foi explorada a utilização de um drone *Parrot AR 2* em conjunto de um computador para realizar detecção de objetos. Os autores afirmam na conclusão que “os resultados demonstraram que, ao utilizar a arquitetura SSD para detecção de objetos com uma única classe, é possível reduzir significativamente o tempo computacional sem comprometer a acurácia. Especificamente, o sistema atingiu uma eficiência de detecção de 98% e, ao integrar o rastreamento do objeto alvo com controladores PID simples, alcançando uma precisão de 96,5% no plano tridimensional (x, y e z)”.

Em "*Object Detection from the Video Taken by Drone via Convolutional Neural Networks*" por *Chenfan Sun et al*, outro trabalho que se destaca, os autores exploraram o uso de CNNs para permitir que drones reconheçam objetos como edifícios, carros, árvores e pessoas, utilizando aprendizado por transferência e a API de detecção de objetos do TensorFlow. Esse estudo comparou os detectores SSD e Faster R-CNN, constatando que o SSD é mais rápido (115 ms por quadro) porém menos preciso, enquanto o Faster R-CNN é mais preciso (95% de detecção) entretanto mais lento (140 ms por quadro). Para reduzir o tempo de execução do Faster R-CNN, os autores sugeriram limitar o número de regiões propostas. Além disso, foi analisado o consumo de memória dos modelos MobileNet, ResNet50 e GoogleNet/Inception V2, com o MobileNet sendo o mais eficiente (menos de 1 GB) e o ResNet50 o mais exigente (cerca de 5 GB). O estudo destacou a relação entre tempo de execução, consumo de memória e extratores de características.

Tais projetos também usam equipamentos especializados com hardware não-modular o que dificulta futuras atualização mudando só a FCU ou o embarcado. Então o projeto se propôs a montar uma plataforma com os recursos possíveis que seria de fácil manutenção, melhoria e utilização.

É importante levar em conta que existem trabalhos propondo usar identificação de objetos usando computadores embarcados e processando as informações localmente ou enviando as imagens a uma estação de chão. Além disso, a realização

de competições de melhora de algoritmos para identificação usando imagens de drones.

Contudo, o projeto visava uma implementação usando peças mais disponíveis e com maior facilidade de realização do retreinamento de modelos, sendo YOLO ou outros, que poderiam ser usados para propósitos distintos e em áreas sem, necessariamente, conexão à internet.

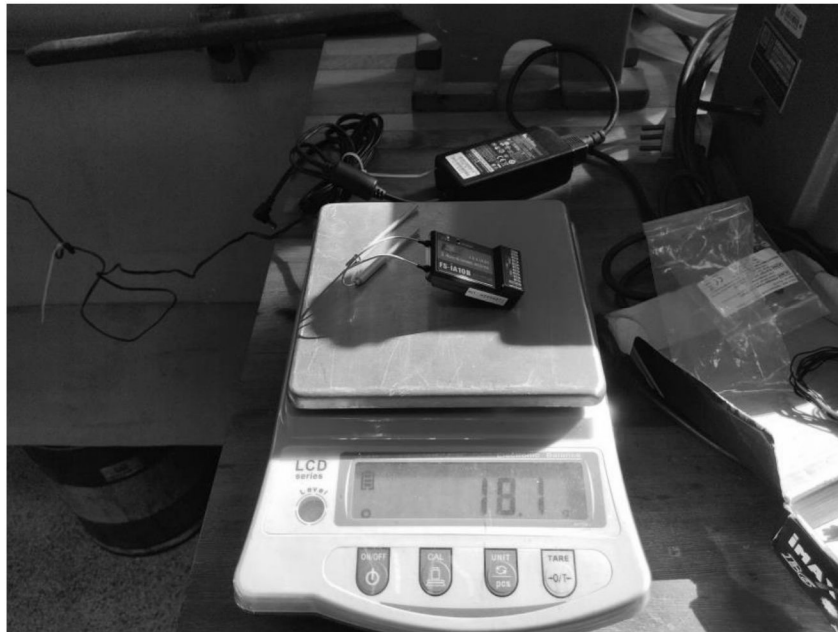
3 METODOLOGIA

3.1 Testes em Bancada

Na execução do projeto foram realizados dois testes de bancada: teste de continuidade e de medição da massa dos componentes. Foram utilizados, respectivamente, um multímetro digital e uma balança digital calibrada e ajustada.

Para a medição da massa, cada elemento que fazia parte do drone foi colocado na balança para obter a massa de cada um deles. Tais valores foram obtidos usando uma quantidade conhecida de objetos e dividindo a massa resultante pelo seu número. Os resultados foram expostos em forma de uma tabela (Tabela 1) que contém a massa e uma fórmula que efetuou a conta da massa do drone em si. Como resultado foi obtido um peso base de 1898,5 g ou 1,9 kg o que entra na classe 3 de drones. Uma imagem do teste realizado está na Figura 14.

Figura 14 - Exemplo da medição da massa.



Fonte: Autor, 2024.

Tabela 1 - Resultados da medição de massa

COMPONENTE	QTDE	Massa(g)	TOTAL
<i>PIXHAWK 6C</i>	1	58,4	58,4
<i>GPS M8N</i>	1	32,5	32,5
TELEMETRIA SIK 433MHz	1	15,3	15,3
PM02	1	20,0	20,0
Placa de Distribuição	1	21,9	21,9
<i>Frame</i>	1	802	802
Bateria	1	544	544
Conversor EC5 para XT60	1	10	10
<i>UBEC</i>	1	40	40
Câmera SIYI A2	1	85	85
Cabos	1	60	60
Suportes adicionais	1	30	30
Jetson Nano	1	100	100
Receptor Radio	1	18,1	18,1
Placa de conexão PWM	1	11,8	11,8
Total			1959

Fonte: Autor, 2024.

Para verificar a conexão de cada componente adaptado, fez-se necessário fazer o teste de continuidade. O teste de continuidade se faz usando um multímetro com funcionalidade de continuidade. Primeiro, o equipamento não pode estar alimentado durante o teste. Segundo, são usadas as pontas de prova no modo de teste de tensão para fazer uma conexão entre os pontos da extremidade de cada conexão. Se houvesse continuidade, haveria um *beep* do multímetro. Esse teste, além de ter verificado a continuidade entre as partes que deveriam estar conectadas, também identificou possíveis falhas que poderiam ocasionar um acidente no futuro.

3.2 Montagem

3.2.1 Montagem Básica

A Montagem básica do quadricóptero foi realizada usando um kit criado pela Holybro chamado X500 V2. Foi adquirido o kit do frame com alguns acessórios inclusos. Isso ocorreu pelo fato de que, já tinha sido realizada a compra de alguns componentes do kit separadamente, como os motores, ESCs e hélices. O kit pode ser visto na figura 15.

Figura 15 - Kit Frame X500 V2 em sua caixa



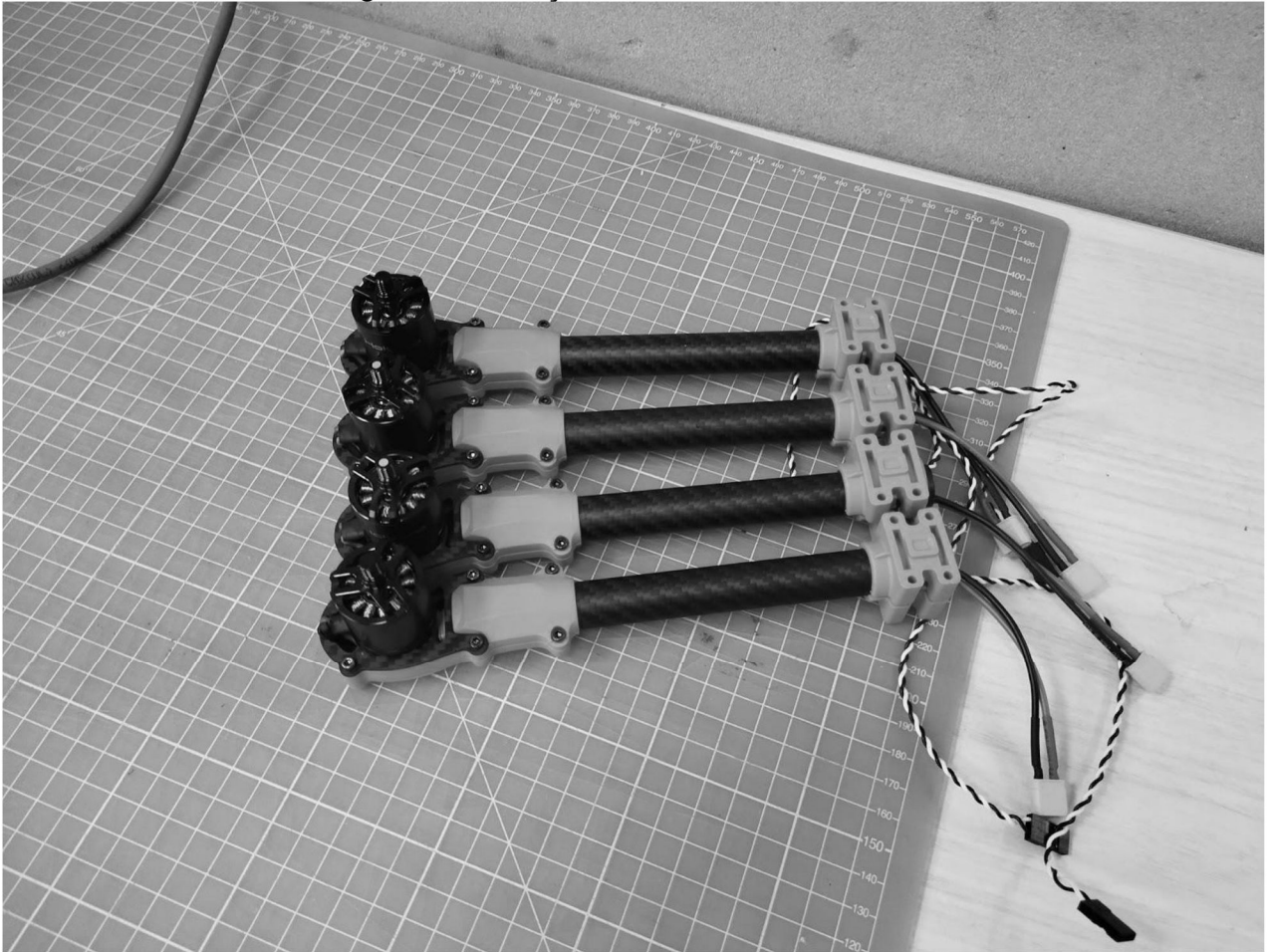
Fonte: Autor, 2024.

Primeiramente, foram montados os braços do drone e depois foi montada a parte inferior da estrutura do *frame*. Em seguida, foram retirados os componentes que foram usados no drone anterior. Em específico, a *pixhawk*, *GPS* e módulo *wireless* para conexão com a estação de chão. Essas partes foram juntadas e montadas em seguida para obter um drone básico sem ter a *Raspberry Pi* ou a *Jetson Nano*

embarcada. Para dar os próximos passos de integração com a *Raspberry* e câmera, houve a necessidade de configurar o básico do drone.

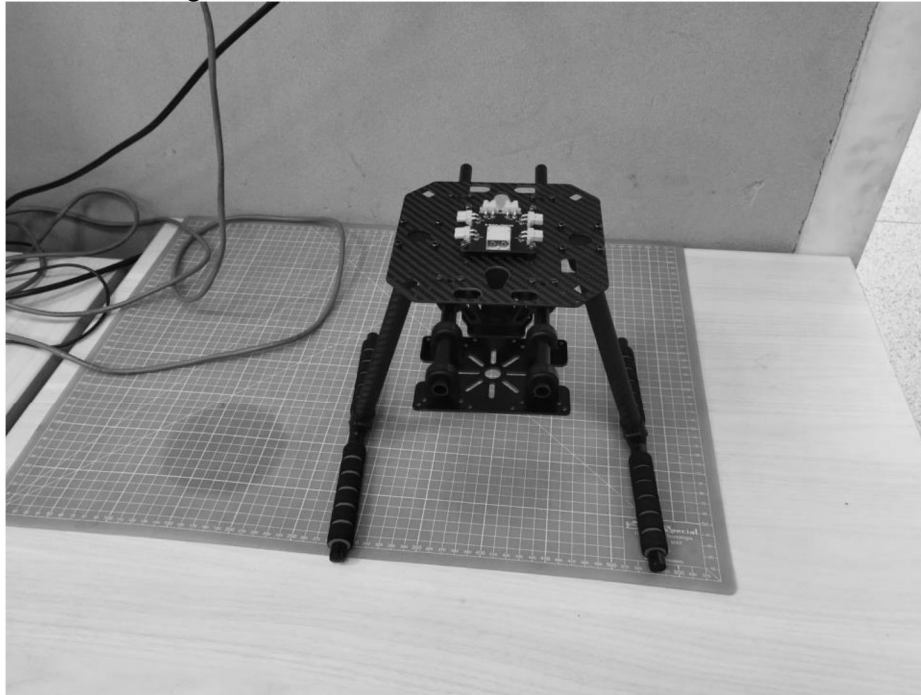
Na plataforma superior, posicionou o *FCU (Pixhawk)*, *GPS*, adaptador para motores e módulo de comunicação *wireless*. Da mesma forma, na plataforma inferior, ficou apenas o *Power Module 02* e a placa distribuidora do drone. Entre a bateria e a placa inferior foi colocado o rádio receptor.

Figura 16 - Braços do drone montados



Fonte: Autor, 2024.

Figura 17 - Parte inferior do frame montada



Fonte: Autor, 2024.

Figura 18 - Drone básico montado em meio de teste de sentido dos motores



Fonte: Autor, 2024.

Figura 19 - Imagem do protótipo



Fonte: Autor, 2025.

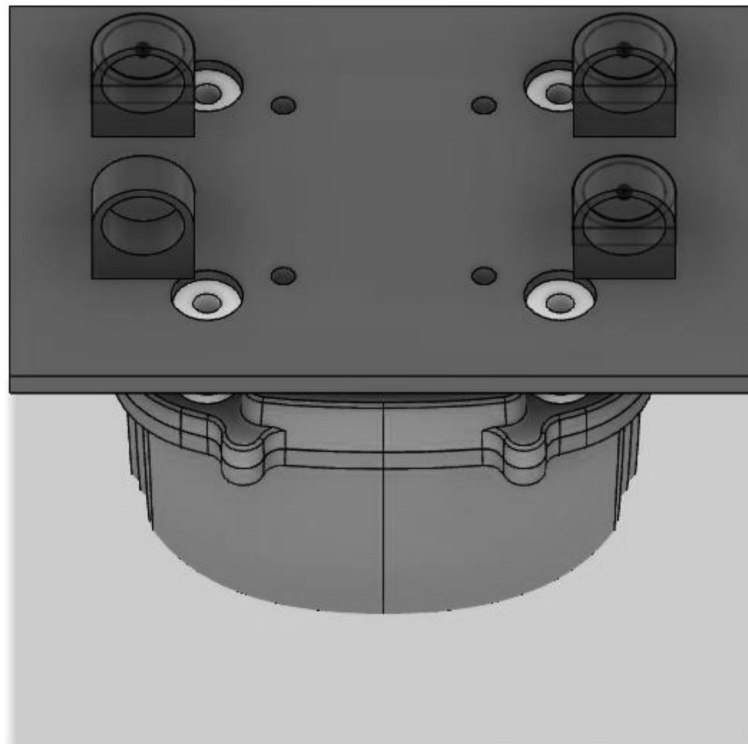
3.2.2 Modelagem dos Suportes

Mostrou-se necessário, durante o processo, a modelagem de um suporte que possuísse furos compatíveis com os da câmera. Foi elaborado um suporte inicial feito de MDF e peças impressas, utilizado para fins de testes antes de instalar a câmera no drone.

Para esse fim, utilizou-se a documentação do fabricante para projetar os furos em um modelo CAD, uma impressora 3D para imprimir os conectores, uma cortadora a laser para cortar a placa de MDF de 3mm e um software para desenvolver os modelos. Para a *Raspberry Pi*, foi usado um suporte encontrado e impresso para ser usado no drone.

A modelagem foi realizada no software *FUSION 360*. Os resultados foram apresentados na figura 20.

Figura 20 - Suporte para a câmera Siyi A2 Mini



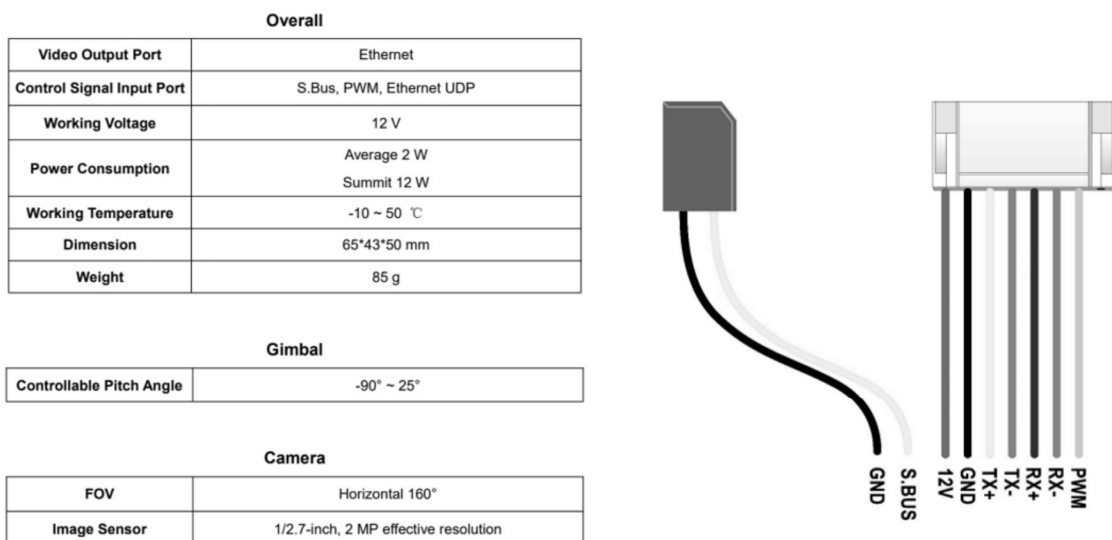
Fonte: Autor, 2024.

3.2.3 Adaptação da Câmera SIYI A2 Mini

O conector RJ45 se trata do tipo usado para conexão pela internet cabeada pela maioria dos computadores. Normalmente, chamam de *Ethernet Port*. A SIYI A2 mini vem com conectores JST GH pois são os usados nos equipamentos da SIYI. Porém, no manual da empresa fabricante, existe uma parte que explica como a câmera consegue se comunicar com dispositivos que não são da SIYI. Para tal feito, foi necessária uma adaptação do GH para RJ45 pois é o conector que é usado no equipamento que seria conectado à câmera.

Estão expostas na figura 21 algumas informações do manual do equipamento, em específico, o *pinout* dos cabos e o tipo de protocolo para comunicação e visualização das imagens obtidas pela câmera.

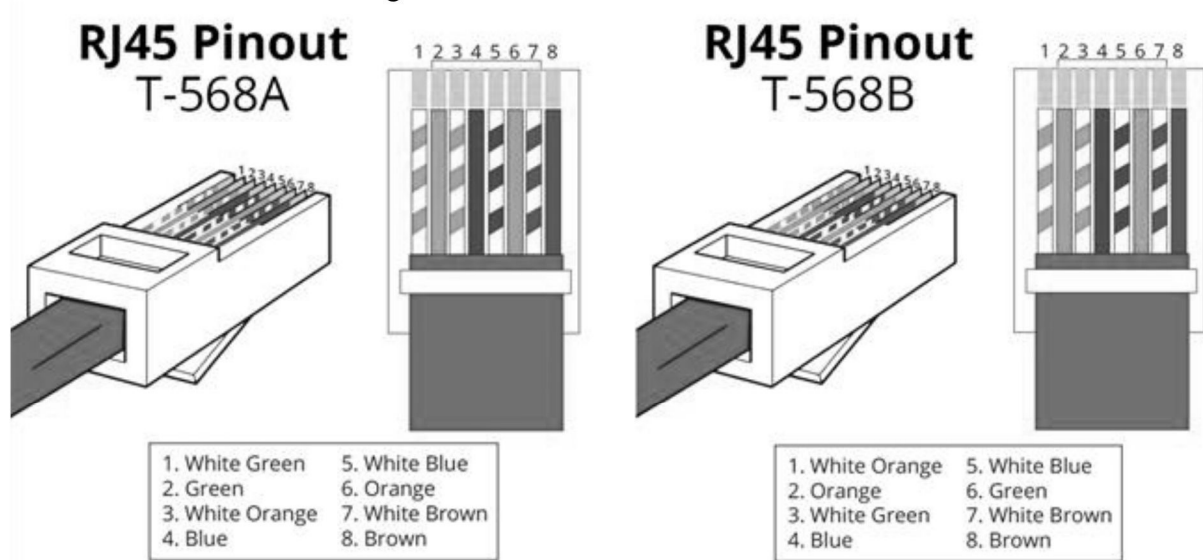
Figura 21 - Pinout dos conectores e especificações da câmera Siyi A2 Mini



Fonte: Adaptado de Siyi, 2024.

A câmera consegue ser controlada via protocolo UDP e *SDK* da fabricante. Porém, o vídeo pode ser acessado via protocolo RTSP pela porta Ethernet do dispositivo. Contudo, a conexão devia ser do tipo *straight-through* e não *crossover* pois havia probabilidade de danos no equipamento. Foi usado um RJ45 T-568B.

Figura 22 - Pinout do conector RJ45



Fonte: BDFIX, 2018.

Foi necessário seguir o diagrama na figura 22 pois haveria necessidade de usar os quatro cabos (RX+, RX-, TX+ e TX-) nos pinos designados na tabela 2.

Então, foi feita a conexão com os seguintes pinos demonstrados na tabela seguinte:

Tabela 2 - Relação da conexão necessária para a adaptação

Pino RJ45	Cor do cabo no conector RJ45	Siyi A2 mini
1	Branco-Laranja	TX+
2	Laranja	TX-
3	Branco-Verde	RX+
4	Azul	
5	Branco-Azul	
6	Verde	RX-
7	Branco-Marrom	
8	Marrom	

Fonte: Autor, 2024.

Sendo que o VCC e GND foram conectados no UBEC com saída 12V e o SBUS e segundo GND não foram conectados.

Foi possível, encontrar a solução para um erro de comunicação entre a câmera e a placa embarcada com modificações no dnsmasq e dhcp da *Raspberry Pi* 5B e *Jetson Nano*. O código para configuração da comunicação entre a câmera foi disponibilizado no github da equipe.

3.3 Configuração

3.3.1 Configurações Básicas no Mission Planner

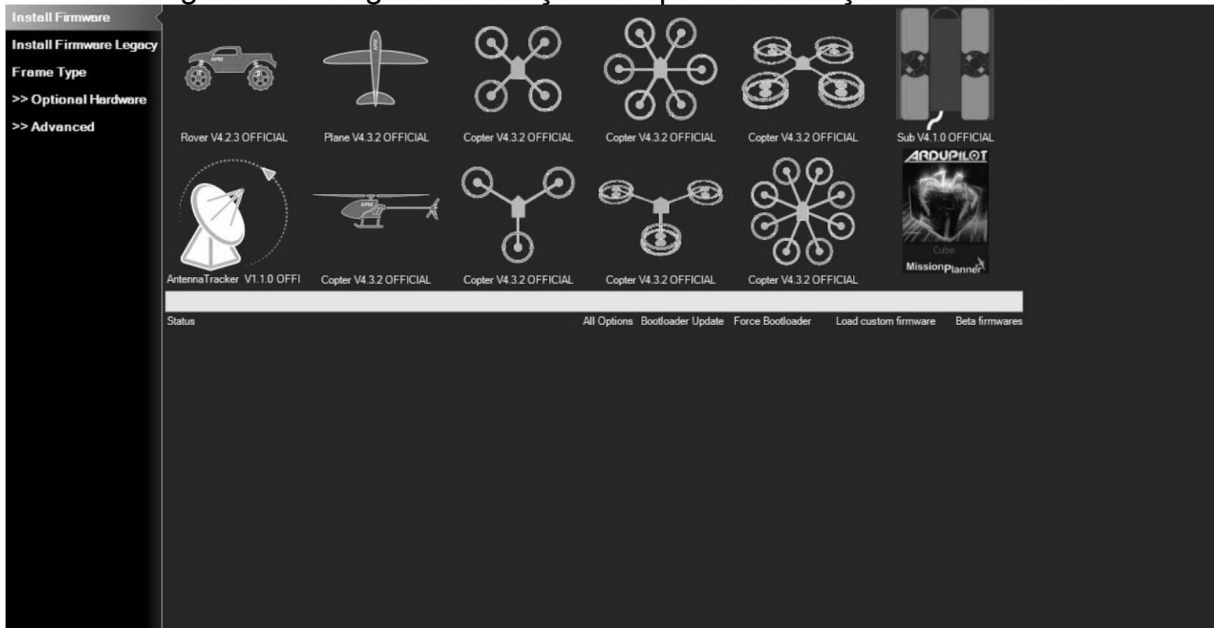
Para a configuração básica do drone, primeiro é obrigatório fazer a instalação do firmware do tipo de drone que ainda será feito. A instalação é realizada conectando a controladora no software *Mission Planner* e indo na aba de *Setup* e na categoria instalar firmware onde pode escolher o tipo de firmware a ser instalado.

Depois de instalado, se conecta a controladora usando o botão de *connect* e selecionando a porta que está conectada à mesma. Primeiramente, é necessário calibrar o tipo de *frame* do drone, calibrar os acelerômetros do drone, calibrar o *compass* do drone e testar os motores para verificar se estão em ordem e sentido certos.

Os acelerômetros são calibrados mudando a posição do drone para a que é dada pelo software durante a calibragem. Para calibrar as bússolas, deve-se girar o drone nos eixos *Roll*, *Pitch* e *Yaw* do drone até que a barra de calibragem esteja cheia. Então a calibragem começa quando se clica no botão e depois a controladora deve ser reiniciada clicando-se no botão *reboot*. Para o teste dos motores, coloca-se um valor em porcentagem de potência dos motores onde eles giram levemente. Após isso é necessário checar os motores um por um na sequência alfabética. Para verificação do sentido de rotação usa-se um esquemático que mostra os sentidos dos motores no tipo de frame utilizado.

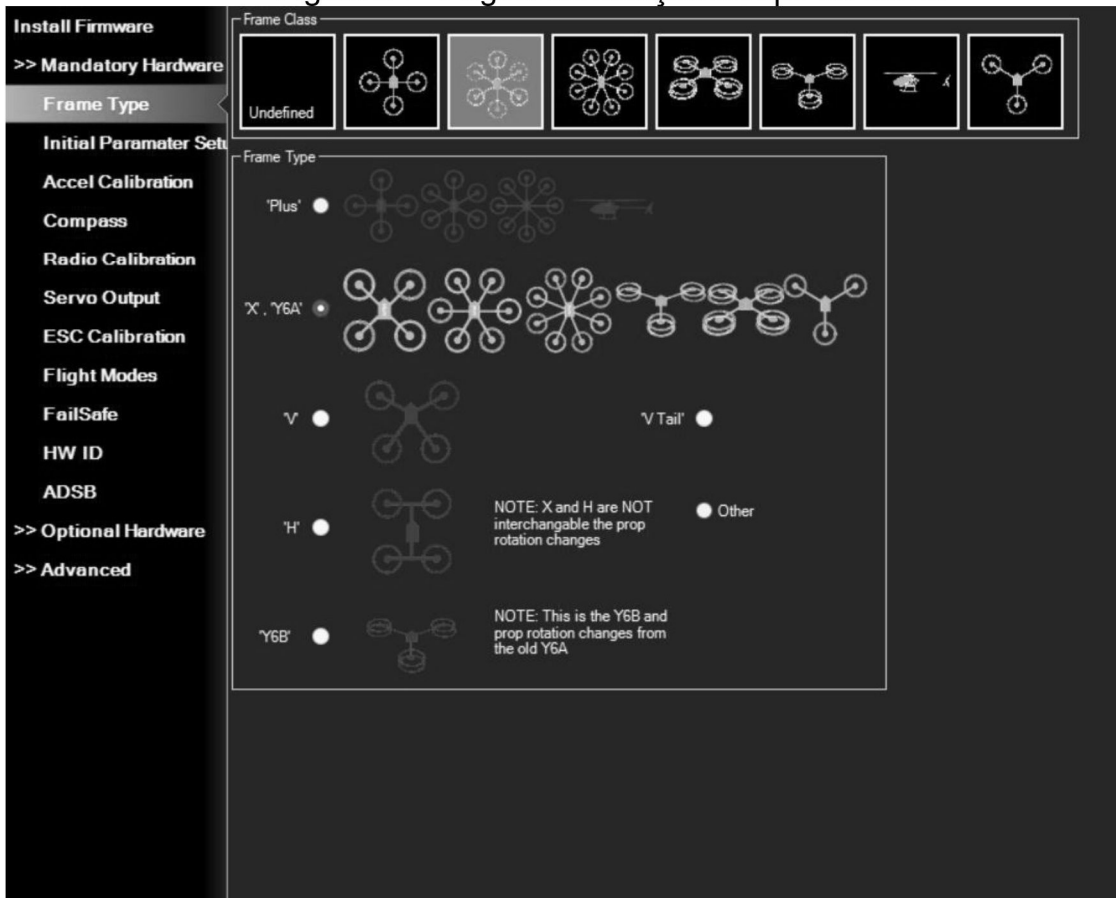
Depois, deve ser feito o *initial parameter setting*, na aba de mesmo nome, preenchendo os valores do tamanho das hélices (*airscrew size*), valores mínimos e máximos da bateria, o tipo da mesma e a quantidade de células da bateria que será utilizada. Em seguida, pressiona-se o botão de *Calculate Initial Parameters*. Esse procedimento dará os valores iniciais de PID. As páginas de calibração estão sendo mostradas nas figuras 23 a 28.

Figura 23 - Página de seleção do tipo de instalação do firmware



Fonte: Autor, 2024.

Figura 24 - Página de seleção do tipo de frame



Fonte: Autor, 2024.

Figura 25 - Página de calibração do compass

Compass Priority

Set the Compass Priority by reordering the compasses in the table below (Highest at the top)

Priority	DevID	Bus Type	Bus	Address	DevType	Missing	External	Oerntation	Up	Down
1	658953	I2C	1	14	IST8310	<input checked="" type="checkbox"/>	<input type="checkbox"/>		↑	↓
2	658433	I2C	0	12	IST8310	<input type="checkbox"/>	<input type="checkbox"/>	None	↑	↓

Do you want to disable any of the first 3 compasses?
 Use Compass 1 Use Compass 2 Use Compass 3 Automatically learn offsets

A reboot is required to adjust the ordering.

A mag calibration is required to remap the above changes.

Onboard Mag Calibration

Mag 1

Mag 2

Mag 3

Fitness Relax fitness if calibration fails

Large Vehicle MagCal

Fonte: Autor, 2024.

Figura 26 - Página de calibração do acelerômetro

Accelerometer Calibration

Level your Autopilot to set default accelerometer Min/Max (3 axis).
 This will ask you to place your autopilot on each edge.

Level your Autopilot to set default accelerometer offsets (1 axis/AHRS trims).
 This requires you to place your autopilot flat and level.

Level your Autopilot to set default accelerometer scale factors for level flight (1 axis).
 This requires you to place your autopilot flat and level.

Fonte: Autor, 2024.

Figura 27 - Página de teste dos motores



Fonte: Autor, 2024.

Figura 28 - Página de calibração parâmetros iniciais



Fonte: Autor, 2024.

3.3.2 Escolha da Placa a Ser Embarcada

Inicialmente, seria usada uma placa de desenvolvimento *Jetson Nano* 4gb ram para ser embarcada no drone, entretanto, esta não possui mais apoio para atualizações e nem é mais fabricada pela Nvidia, o que não permitia o uso da biblioteca *python ultralytics* sem adaptações. Porém, durante o período do projeto, foi lançada a placa *Raspberry Pi 5B* de menor dimensão, mais leve e oferecendo mais apoio da fabricante e comunidade.

Com o anúncio e subsequente venda do Kit AI e do AI Hats feitos pela *Raspberry Pi Foundation* e *Hailo*, foi tomada a decisão como equipe de trocar a placa embarcada no drone. O Kit IA foi um sistema de desenvolvimento fabricado e distribuído pela *Raspberry Pi Foundation* e *Hailo*, ele consistia em uma *NPU* da *Hailo*, a *Hailo-8L* de 13 TOPS e um *Hat m.2* para *Raspberry 5*. Em quesito de IA, o novo embarcado, em papel, era muito melhor que a *Jetson* pelo formato e desempenho muito superior. Ele não é mais comercializado pois foi criado o *Hat* com a *NPU* embutida, que funciona da mesma maneira. O processo de troca foi relativamente rápido depois de ser enviada a placa e o *HAT* com a *NPU*.

Após um mês da aquisição, foi possível constatar nos dados disponibilizados pelo *data root labs* que, pelo desempenho do AI kit em relação à *Jetson Nano*, que foi acertada a escolha da mudança de plataforma.

3.3.3 Calibração e Ajuste do PID

Para a calibração e ajuste do PID, existem alguns tipos de técnicas que podem ser usadas, sendo elas, o ajuste manual com uma bancada, o ajuste usando *AUTOTUNE* e o script LUA de *quiktune*. O ajuste manual precisa de uma bancada específica para a realização pois é necessário que ele pare de forma uniforme e seja feita a simulação de rajadas empurrando o drone em um de seus eixos. De acordo com a reação da aeronave, são feitos ajustes até encontrar valores onde a mesma não se posicione de maneira muito agressiva nem perca controle. Tal processo é feito de maneira automática com o *AUTOTUNE* e *quiktune* com algumas diferenças. Durante o *AUTOTUNE*, o drone pode ficar ou em *ALTHOLD* onde ele irá vagar, porém manter a altura ou pode ficar em *LOITER* onde o drone tentará ficar na mesma posição. O *ALTHOLD* é melhor no caso pois os valores ficam mais precisos. O *quiktune* por sua vez pode somente ser feito em *LOITER* e ele não calcula todos os valores do PID em especial os da aceleração.

Em razão disso, para o projeto foi escolhido o *quiktune* por ser mais rápido e ter um ajuste razoável.

3.3.4 Modificação do código para uso no drone

Para realização dos testes com o novo computador embarcado, foi necessário modificar programas e códigos para fazer tanto a câmera quanto a *NPU* funcionarem como o projeto demandava. Primeiramente, foram adquiridos os scripts *python* da Hailo para desenvolvimento de aplicações usando sua *NPU* que foram posteriormente modificados para funcionar usando câmeras RTSP. Em seguida, foram adicionadas flags para salvamento de vídeo e salvamento de frames com detecções de pessoas usando a flag `--save` e `-u` respectivamente. A parte do salvamento de vídeo foi feita usando o tutorial do *data root labs*.

Como ocorreram algumas modificações no código da Hailo e foram utilizadas versões mais antigas, para reproduzir o funcionamento, é necessário baixar os códigos modificados diretamente do GitHub da equipe. Esses códigos, junto com outros experimentais, estão disponíveis no repositório do GitHub da equipe.

Alguns trechos de código podem ser vistos nas figuras 29 e 30.

Figura 29 - Parte do código para controle do gimbal da SIYI A2 Mini

```

current = os.path.dirname(os.path.realpath(__file__))
parent_directory = os.path.dirname(current)

sys.path.append(parent_directory)

from siyi_sdk import SIYISDK

master = mavutil.mavlink_connection('/dev/ttyAMA0', baud=115200)
master.wait_heartbeat()
print("Conectado ao sistema:", master.target_system, ", componente:", master.target_component)

def test():
    cam = SIYISDK(server_ip="192.168.144.25", port=37260)
    if not cam.connect():
        print("No connection ")
        exit(1)

    cam.setGimbalRotation(0, -25) # de -90 ate 25 graus.
    cam.disconnect()

```

Fonte: Autor, 2024.

Figura 30 - Parte do código para registrar a localização e quantidade de pessoas na foto em arquivo JSON

```

class user_app_callback_class(app_callback_class):
    def __init__(self):
        super().__init__()
        self.new_variable = 42
        self.frame_count = 0
        self.detection_data = [] # list to hold detection data for each frame
        self.gps_coordinates = (0, 0, 0) # to hold latitude, longitude, altitude

    def gsmallink(self):
        msg = mavutil.mavlink_message('GLOBAL_POSITION_INT')
        if msg and msg.get_type() == 'GLOBAL_POSITION_INT':
            lat = msg.lat / 1E7 # Converte para graus
            lon = msg.lon / 1E7 # Converte para graus
            alt = msg.alt / 1000.0 # Converte para metros
            self.gps_coordinates = (lat, lon, alt)
            print(f"Coordenadas GPS - Latitude: {lat}, Longitude: {lon}, Altitude: {alt} m")
            return self.gps_coordinates

    def increment(self):
        self.frame_count += 1

    def get_count(self):
        return self.frame_count

    def add_detection_data(self, frame_number, detection_count, coordinates):
        self.detection_data.append({
            'frame': frame_number,
            'number_of_people': detection_count,
            'coordinates': coordinates
        })

    def save_to_json(self, filename="detections.json"):
        with open(filename, "w") as f:
            json.dump(self.detection_data, f, indent=4)

    def app_callback(self, info, user_data):
        buffer = info.get_buffer()
        if buffer is None:
            return Gst.PadProbedStatus.OK

        user_data.increment()
        frame_number = user_data.get_count()
        user_data.gsmallink() # atualiza as coordenadas GPS
        coordinates = user_data.gps_coordinates

        format, width, height = gst_caps_from_pad(pad)

        frame = None
        if user_data.use_frame and format is not None and width is not None and height is not None:
            frame = gst_map_from_buffer(buffer, format, width, height)

        roi = halo.get_roi_from_buffer(buffer)
        detections = roi.get_objects_type(halo.HALLO_DETECTION)

        detection_count = 0
        person_detected = False
        for detection in detections:
            label = detection.get_label()
            confidence = detection.get_confidence()
            if label == "person":
                detection_count += 1
                person_detected = True

        if user_data.use_frame:
            cv2.putText(frame, f"Detections: {detection_count}", (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
            cv2.putText(frame, f"GPS: {coordinates[0]:.3f}, {coordinates[1]:.3f}", (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)

            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            user_data.set_frame(frame)

            if person_detected:
                image_dir = "imagens"
                os.makedirs(image_dir, exist_ok=True)
                image_path = os.path.join(image_dir, f"frame_{frame_number}.jpg")
                cv2.imwrite(image_path, frame)

        user_data.add_detection_data(frame_number, detection_count, coordinates)
        print(f"Frame: {frame_number}, Detections: {detection_count}, Coordinates: {coordinates}")
        user_data.save_to_json()

```

Fonte: Autor, 2024.

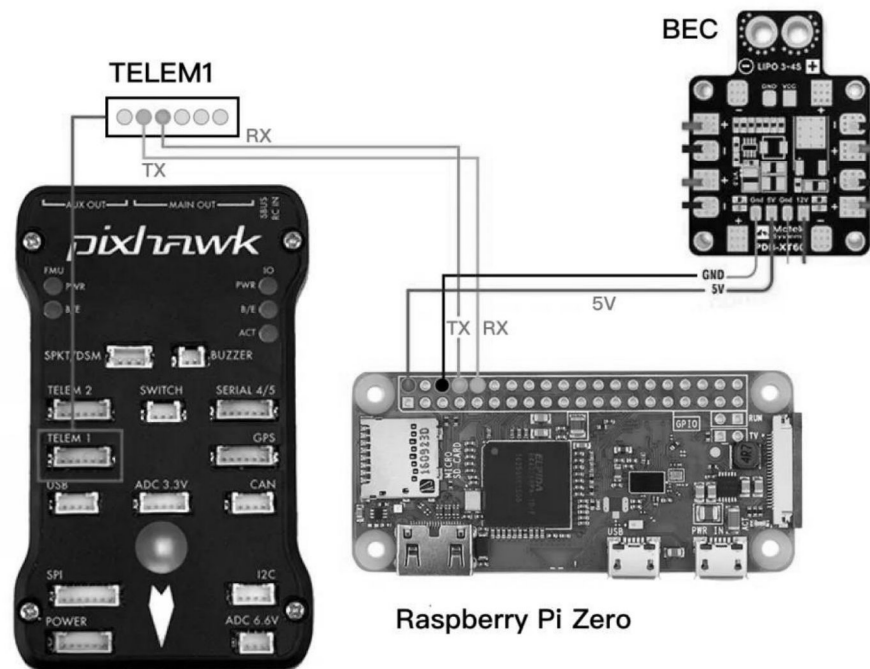
3.3.5 Comunicação Entre Raspberry Pi/Jetson Nano e Pixhawk

Para que houvesse comunicação entre a *Raspberry Pi* e a *Pixhawk* via protocolo *MAVLink*, foi feita uma conexão via protocolo UART. A porta responsável na nossa controladora foi a Telem2. Para alimentação para a *Raspberry*, os pinos 5 VCC e GND também foram necessários, contudo, como o consumo de corrente da *Raspberry* pode ser alto, é importante alimentá-la via UBEC Externo. Assim, foi feita uma adaptação de forma que o GPIO se ligou a tal porta, como mostrado na figura 31.

A configuração de software para a comunicação adequada exige a habilitação da porta UART via *raspi-config* no *rasbian* e alteração dos parâmetros no controlador de voo. São eles:

`SERIAL2_PROTOCOL = 2` (o padrão) para ativar o *MAVLink 2* na porta serial;
`SERIAL2_BAUD = 115` para que o controlador de voo possa se comunicar com a *Raspberry* em 115200 *baud*.

Figura 31 - Conexão UART entre Raspberry Pi e Pixhawk



Fonte: Bellergy, 2020.

Um detalhe importante sobre tal configuração é que podem aparecer mensagens com as configurações quando aberto o *mavproxy* usado para a verificação da conexão. Isso pode ser considerado normal.

3.3.6 Retreinamento de modelos YOLO

Durante o projeto, foram exploradas diferentes abordagens para o treinamento de modelos YOLO. Inicialmente, o modelo utilizado nos testes foi apenas uma conversão direta dos modelos YOLO originais da *Ultralytics* para o formato HEF. Esses modelos foram previamente treinados com o *dataset MS COCO*, o que os tornava menos adequados para a identificação dos objetos específicos desejados em cenários similares aos capturados pela câmera embarcada.

Para melhorar a compatibilidade do modelo com o contexto do projeto, a primeira tentativa de treinamento foi realizada com o *dataset People on Grass*. No entanto, os resultados das imagens de validação indicaram baixa precisão, conforme ilustrado na Figura 32. Em seguida, foi utilizada a plataforma *Roboflow* para a criação de um novo *dataset*, combinando imagens de diferentes fontes. Entretanto, devido ao grande aumento no tamanho do *dataset*, tornou-se inviável realizar o treinamento localmente em uma máquina equipada com um *Ryzen 5600G*, 16 GB de memória *RAM DDR4* e uma *RTX 3060*. Como alternativa, foi utilizada a ferramenta de treinamento da própria plataforma *Roboflow*.

Apesar do sucesso no retreinamento, a integração do novo modelo ao protótipo enfrentou dificuldades devido a incompatibilidades entre as versões das bibliotecas e ferramentas de desenvolvimento do *Hailo* e a versão instalada na *Raspberry Pi*. Como resultado, não foi possível utilizar o modelo retreinado no sistema.

4 RESULTADOS E DISCUSSÕES





























O projeto começou com a especificação e aquisição dos componentes para um drone protótipo. Tal drone necessitaria ter um *payload* compatível com o peso dos equipamentos que seriam embarcados no drone, a câmera e o computador embarcado. Foi escolhido pelo fator de confiabilidade, a compra e uso do KIT de desenvolvimento X500V2 da Holybro com peças adquiríveis para troca e um *payload* de 1kg adicional.

O computador embarcado seria uma *Jetson Nano* inicialmente, porém como já exposto, foi modificada a plataforma para a placa *Raspberry Pi 5B* com o KIT AI. Como foi adquirida em prazo próximo do lançamento, a plataforma apresentou alguns problemas com a importação e utilização de modelos não feitos pela própria *Hailo*. Apesar de serem feitos testes com o método de importação de modelos YOLOv10 da *ultralytics* mostrados no github e em alguns outros sites, não foi possível sua utilização. A razão foi a falta de um arquivo ou função de pós processamento customizada no código disponibilizado pela fabricante. Apesar de ter mudado de plataforma, ainda foi possível usar as duas para reconhecimento usando YOLOv8 e câmeras RTSP e CSI.

Sobre o código, foram usadas como base alguns códigos tanto de controle do *gimbal* da câmera quanto as pipelines básicas da *Hailo*. Foram feitas modificações para reconhecer e usar câmeras RTSP, salvar o vídeo da câmera, salvar o tempo e posição do drone em um arquivo JSON e salvar os frames com a data, horário e posição do drone.

Com as ferramentas do Roboflow, foi possível testar como ficaria um modelo retreinado usando o dataset modificado e foi possível perceber que as medias de mAP, precisão e recall diminuía com as modificações no mesmo, como podemos ver na figura 32.

Figura 32 - Quantidade de modelos e modificações no dataset realizados

MODEL NAME	UPDATED	METRICS	TYPE	DATASET VERSION
 DroneHumanView 8 ID: dronehumanview-9ckcp/8 	 03/09/2024, 13:34	mAP 18.8%  Precision 78.5%  Recall 16.7% 	Roboflow 3.0 Object Detection (Fast)	2024-09-03 2:54pm 
 DroneHumanView 6 ID: dronehumanview-9ckcp/6 	 23/08/2024, 08:40	mAP 20.4%  Precision 50.6%  Recall 10.2% 	YOLO-NAS Object Detection (Accurate)	2024-08-23 3:55am 
 DroneHumanView 4 ID: dronehumanview-9ckcp/4 	 16/08/2024, 05:36	mAP 79.8%  Precision 86.5%  Recall 73.1% 	Roboflow 3.0 Object Detection (Fast)	2024-08-16 1:09am 
 DroneHumanView 3 ID: dronehumanview-9ckcp/3 	 15/08/2024, 23:48	mAP 74.5%  Precision 79.5%  Recall 67.6% 	yolov8s Model Upload	2024-08-11 3:20am 

Fonte: Autor

Ademais, foi possível, realizar voos para testes do conjunto antes de colocar o embarcado e câmera. Durante os voos para calibração e ajuste do conjunto, foram logrados voos bem longos, com duração de 10-15 minutos, e verificou-se que o conjunto de controle de voo estava funcionando muito bem. Durante os voos com o protótipo já pronto com uma duração menor de 7-8 minutos, entretanto com tudo já funcionando e com baterias à 4.0V. Além disso, foi possível fazer alguns vídeos e imagens dos voos.

Apesar dessas dificuldades, foi possível adquirir resultados e realizar alguns testes antes do período de chuva, no qual não é possível voar com o drone pela imprevisibilidade do clima e problemas de rajadas entre outros.

Nas figuras 33 e 34, é possível ver os resultados da utilização do drone tanto no período matutino quanto ao final da tarde com condições de alta e baixa luminosidade respectivamente.

Figura 33 - Foto de uma das filmagens com o YOLOv8s em baixa luminosidade



Fonte: Autor, 2024.

Figura 34 - Foto de uma das filmagens com o YOLOv8s em alta luminosidade



Fonte: Autor, 2024.

5 CONCLUSÃO

Ao final da execução do projeto, o autor reconheceu os resultados obtidos como altamente satisfatórios, evidenciando o cumprimento dos objetivos estabelecidos. Durante a primeira etapa, foi possível realizar a montagem do drone, testá-lo em voo com sucesso e implementar um programa de reconhecimento de objetos utilizando o modelo YOLO na *Jetson Nano*. Essa fase inicial demonstrou a viabilidade técnica do projeto e estabeleceu uma base sólida para os avanços subsequentes.

Na segunda etapa, o projeto evoluiu significativamente com a substituição da *Jetson Nano* pela *Raspberry Pi*, o que demandou ajustes nos scripts e otimizações para garantir a compatibilidade com o novo hardware. Além disso, foram realizados testes rigorosos com o drone modificado, corrigindo os problemas identificados anteriormente e aprimorando o desempenho do sistema. Esses ajustes resultaram em uma solução mais robusta e eficiente, alinhada às expectativas iniciais.

O período dedicado ao projeto foi uma experiência de aprendizado intenso, proporcionando ao autor a oportunidade de aplicar conhecimentos teóricos em situações práticas e desafiadoras. O trabalho não apenas permitiu explorar novas tecnologias, mas também fomentou a integração e cooperação entre profissionais e estudantes de diferentes áreas dentro do campus, enriquecendo o projeto com perspectivas multidisciplinares.

Para a continuidade, seria extremamente benéfico que outros projetos explorassem a infraestrutura e o hardware desenvolvidos, seja para o aprimoramento do software existente, seja para a aplicação do drone em novos cenários. Por exemplo, o drone poderia ser adaptado para operações específicas, como monitoramento ambiental, suporte a resgates ou automação em áreas agrícolas. Tais iniciativas não apenas ampliariam o impacto do trabalho realizado, mas também contribuiriam para o desenvolvimento tecnológico e a formação de novas soluções inovadoras. Ou talvez, um projeto que procure o aprimoramento da parte do treinamento e integração do modelo retreinado na *Raspberry Pi*.

REFERÊNCIAS

ARDUPILOT. **ArduPilot Documentation — ArduPilot documentation**. Disponível em: <<https://ardupilot.org/ardupilot/index.html>>. Acesso em: 14 mar. 2023.

ARDUPILOT. **History of Ardupilot**. Disponível em: <<https://ardupilot.org/dev/docs/common-history-of-ardupilot.html>> Acesso em: 22 de novembro de 2022.

ARDUPILOT. **SimpleBGC Gimbal (aka AlexMos gimbal)**. 2021. Disponível em: <<https://ardupilot.org/copter/docs/common-simplebgc-gimbal.htm>> Acesso em 24 de dezembro de 2022.

AUTERION. **The history of Pixhawk**. Disponível em: <<https://auterion.com/company/the-history-of-pixhawk>> Acesso em: 23 de dezembro de 2022.

BDFIX. **RJ45 Pinout & Wiring Diagrams for Networking**. Disponível em: <<https://bdfixforum.blogspot.com/2016/06/rj45-pinout-wiring-diagrams-for.html>>. Acesso em: 14 mar. 2025.

BELLERGY. **5. Wiring and connecting Pixhawk to Raspberry Pi**. Disponível em: <<https://bellergy.com/5-wring-and-connect-pixhawk-to-raspberry-pi/>>. Acesso em: 14 mar. 2025.

CECCON, D. **Os tipos de redes neurais – IA Expert Academy**. Disponível em: <<https://iaexpert.academy/2020/06/08/os-tipos-de-redes-neurais>>. Acesso em: 13 dec. 2024.

CEZNE, Eric; JUMBERT, Maria Gabrielsen; SANDVIK, Kristin Bergtora. **Drones como veículos para a ação humanitária: perspectivas, oportunidades e desafios**. Revista Conjuntura Austral, Porto Alegre, v.7, n.33-34, p.45-60, dez. 2015/mar. 2016. ISSN: 2178-8839. Disponível em: <<https://seer.ufrgs.br/index.php/ConjunturaAustral/article/view/60267/36712>>. Acesso em: 28 de julho de 2023.

CLOUDFLARE. **O que é uma rede neural?** Disponível em: <<https://www.cloudflare.com/pt-br/learning/ai/what-is-neural-network/>>. Acesso em: 04 jun. 2024.

CONFEDERAÇÃO NACIONAL DO TRANSPORTE. **Drones salvam: conheça os usos positivos da tecnologia**. Disponível em: <<https://cnt.org.br/agencia-cnt/drones-salvam-conheca-usos-positivos-tecnologia>>. Acesso em: 10 dez. 2024.

WONG, C. **CSI Cameras: A Comprehensive Guide to Camera Serial Interface Technology – Supertek**. Disponível em: <<https://www.supertekmodule.com/csi-cameras-a-comprehensive-guide-to-camera-serial-interface-technology/>>.

DELGADO, Sergio Carlavilla. **Tutorial sobre comunicações seriais e UART**. [S.l.]. 2021. Disponível em: <<https://docs.freebsd.org/pt-br/articles/serial-uart>> Acesso em: 20 de dezembro de 2022.

DRL Team AI R&D Center. **AI Kit on a Raspberry Pi 5: setup and running Computer Vision Pipelines**. DataRoot Labs, 19 ago. 2024. Disponível em: <https://datarootlabs.com/blog/hailo-ai-kit-raspberry-pi-5-setup-and-computer-vision-pipelines>. Acesso em: 10 out. 2024.

GUI, A. **Cable Difference: Straight Through vs Crossover Cable**. Disponível em: <<https://www.cables-solutions.com/difference-between-straight-through-and-crossover-cable.html>>

GUIMARÃES, Fábio. **I2C**. 12 de agosto de 2018. Disponível em: <<https://mundoprojetado.com.br/i2c>> Acesso em: 20 de dezembro de 2022.

HAILO-AI. **Hailo Raspberry Pi 5 Examples**. GitHub, 2024. Disponível em: <https://github.com/hailo-ai/hailo-rpi5-examples>. Acesso em: 18 nov. 2024.

HOLYBRO. **Pixhawk 6C**. Holybro. Disponível em: <<http://www.holybro.com/product/pixhawk-6c>> Acesso em: 18 de setembro de 2022.

HUR, Byul et al. Progress in Autonomous Building Inspection Drone Development for Scanning Exterior Damage of Buildings. **The Journal of Management and Engineering Integration**, [S.l.], v. 13, n. 2, p. 28-38, Inverno 2020. Disponível em: <https://www.journalmei.com/_files/ugd/f76c8e_74fa9dcbd0754ce1a46a4826a6964364.pdf#page=28> Acesso em: 25 de dezembro de 2022.

IA EXPERT. **Os tipos de redes neurais**. Disponível em: <https://iaexpert.academy/2020/06/08/os-tipos-de-redes-neurais/?doing_wp_cron=1711060085.8809230327606201171875>. Acesso em: 04 jun. 2024.

IBM. **O que é uma rede neural?** Disponível em: <<https://www.ibm.com/br-pt/topics/neural-networks>>. Acesso em: 04 jun. 2024.

BBC. **Inundações no Rio Grande do Sul: 25 imagens impactantes da tragédia**. Disponível em: <<https://www.bbc.com/portuguese/articles/cx7d1ppl0lzo>>. Acesso em: 13 dec. 2024.

IRAZABAL, Jean-Marc. BLOZIS, Steve. **AN10216-01**. I²C Manual. Estados Unidos: Philips Semiconductors, 2003 Disponível em: <<https://www.nxp.com/docs/en/application-note/AN10216.pdf>> Acesso em: 19 de dezembro de 2022.

KARACA, Y. et al. The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations. **The American Journal of Emergency Medicine**, v. 36, n. 4, p. 583–588, abr. 2018.

KEANE, John F.; CARR, Stephen S. **A brief history of early unmanned aircraft**. Johns Hopkins APL Technical Digest, v. 32, n. 3, p. 558-571, 2013. Disponível

em: <<https://secwww.jhuapl.edu/techdigest/Content/techdigest/pdf/V32-N03/32-03-Keane.pdf>>. Acesso em: 15 jan. 2025.

LENIHAN, A. **O que é o protocolo de transmissão em tempo real (RTSP)**. Disponível em: <<https://www.dacast.com/pt/blog-pr/what-is-real-time-streaming-protocol/>>. Acesso em: 4 jun. 2024.

LYU, M. et al. **Unmanned Aerial Vehicles for Search and Rescue: A Survey**. Remote Sensing, v. 15, n. 13, p. 3266–3266, 25 jun. 2023.

MADEIRA, Daniel. **Protocolo I2C – Comunicação entre Arduinos**. Disponível em : <<https://portal.vidadesilicio.com.br/i2c-comunicacao-entre-arduinos>> Acesso em: 23 de dezembro de 2022.

MANSUR, Rafaela. **Quatro anos da tragédia em Brumadinho: 270 mortes, três desaparecidos e nenhuma punição**. G1, Belo Horizonte, 21 de janeiro de 2023. Disponível em: <<https://g1.globo.com/mg/minas-gerais/noticia/2023/01/25/quatro-anos-da-tragedia-em-brumadinho-270-mortes-tres-desaparecidos-e-nenhuma-punicao.ghtml#2>>. Acesso em: 28 de julho de 2023.

MARTINS, C. **Protocolo RTSP para câmeras IPs**. Disponível em: <<https://aprendacftv.com/protocolo-rtsp-para-cameras-ips/>>. Acesso em: 4 jun. 2024.

MAVLINK. **MAVLink Developer Guide**. 2017. Disponível em: <<https://mavlink.io/en>> Acesso em: 24 de novembro de 2022.

MEDIUM. **YOLO object detection using Opencv with Python**. Disponível em: <<https://medium.com/@MrBam44/yolo-object-detection-using-opencv-with-python-b6386c3d6fc1>> Acesso em: 10 de dezembro de 2022.

MENDONÇA, Helio Souza. **UART**. Disponível em: <<https://paginas.fe.up.pt/~hsm/misc/old/comp/uart/#:~:text=Como%20indicia%20o%20nome%20do,medida%20em%20bits%20por%20segundo>> Acesso em: 20 de dezembro de 2022.

MOTA, S. **Um guia sobre Visão Computacional: Como os computadores enxergam?** Disponível em: <<https://suzana-svm.medium.com/guia-visao-computacional-ae2a2ace0973>>. Acesso em: 14 mar. 2025.

NVIDIA CORPORATION. Guia do Kit de Desenvolvedor NVIDIA® Jetson Nano™: Uma introdução prática à IA e robótica. [S.l.]: NVIDIA Corporation, [2024]. Disponível em: <<https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit>>. Acesso em: 03 jun. 2024.

OBINATA, Hiroyuki. **i7y: AI and Robotics Blog**. i7y, 2024. Disponível em: <<https://i7y.org/en/>>. Acesso em: 10 abr. 2024.

OPENCV. [Site Institucional]. Disponível em: <<https://opencv.org/about>> Acesso em: 20 de dezembro de 2022.

PILOTO POLICIAL. **CBM/DF recebe seis novos drones para operações de resgate - Piloto Policial.** Disponível em: <<https://www.pilotopolicial.com.br/cbmdf-recebe-seis-novos-drones-para-operacoes-de-resgate/>>. Acesso em: 15 jan. 2025.

HOLYBRO. **Pixhawk 6C.** Disponível em: <<https://holybro.com/products/pixhawk-6c>>. Acesso em: 14 mar. 2023.

PIXHAWK. [Site Institucional]. Disponível em: <<https://pixhawk.org>> Acesso em: 22 de dezembro de 2022.

PIXHAWK. **Pixhawk-Standards.**

Q-engineering. **All the machine vision and deep learning solutions.** Q-engineering, 2024. Disponível em: <https://qengineering.eu/>. Acesso em: 18 nov. 2024.

QUAN, Quan. **Introduction to Multicopter Design and Control.** China: 2017.

RASPBERRY PI LTD. **Buy a Raspberry Pi AI Kit – Raspberry Pi.** Disponível em: <<https://www.raspberrypi.com/products/ai-kit/>>. Acesso em: 14 dec. 2024.

RASPBERRY PI. [Site Institucional]. Raspberry Pi. Disponível em: <<https://www.raspberrypi.com>> Acesso em: 26 de dezembro de 2022.

REICHELT. **HOLYBRO Holybro X500 V2 Kit, Drone Kit | Drone kits at very favourable prices | reichelt elektronik.** Disponível em: <<https://www.reichelt.com/be/en/holybro-x500-v2-kit-drone-kit-x500-kit-v2-p324607.html>>. Acesso em: 14 nov. 2024.

RIBEIRO, B. et al. **Barragens que ruíram em Brumadinho e Mariana tinham modelo ultrapassado.** Disponível em: <<https://noticias.uol.com.br/ultimas-noticias/agencia-estado/2019/01/29/barragens-que-ruiram-em-brumadinho-e-mariana-tinham-modelo-ultrapassado.htm>>. Acesso em: 14 mar. 2025.

ROHAN, A.; RABAH, M.; KIM, S.-H. **Convolutional Neural Network-Based Real-Time Object Detection and Tracking for Parrot AR Drone 2.** IEEE Access, v. 7, p. 69575–69584, 2019.

ROHDE&SCHWARTZ. **Compreender UART.** [S.l.]. 2022. Disponível em: <https://www.rohde-schwarz.com/br/produtos/teste-e-medicao/osciloscopios/educational-content/compreender-uart_254524.html> Acesso em: 20 de dezembro de 2022.

ROSEBROCK, Adrian. **Understanding a real-time object detection network: You Only Look Once (YOLOv1).** Disponível em: <<https://pyimagesearch.com/2022/04/11/understanding-a-real-time-object-detection-network-you-only-look-once-yolov1/>>. Acesso em: 04 jun. 2024.

TONMIND. **RTSP: Protocolo de transmissão em tempo real-tonmind.com.** Disponível em: <https://pt.tonmind.com/blog/rtsp-real-time-streaming-protocol_b23>. Acesso em: 14 nov. 2024.

SATOMS. **RJ45 Pinout Wiring Diagram for Ethernet Cat 5, 6 and 7**. Disponível em: <<https://satoms.com/ethernet-cable-pinouts/>>.

SIYI Tecnologia. **A2 mini Ultra Wide Angle FPV Gimbal User Manual**. Disponível em: https://siyi.biz/siyi_file/A2%20mini/A2%20mini%20User%20Manual%20v1.1.pdf. Acesso em: 18/04/2024.

Siyi. **A2 mini Ultra Wide Angle FPV Gimbal User Manual**. Disponível em: <https://siyi.biz/siyi_file/A2%20mini/A2%20mini%20v1.1.pdf>. Acesso em: 14 mar. 2025.

SUN, C. et al. **Object Detection from the Video Taken by Drone via Convolutional Neural Networks**. Mathematical Problems in Engineering, v. 2020, p. e4013647, 13 out. 2020.

TUŚNIO, N.; WRÓBLEWSKI, W. **The Efficiency of Drones Usage for Safety and Rescue Operations in an Open Area: A Case from Poland**. Sustainability, v. 14, n. 1, p. 327, 2021. MDPI AG. Disponível em: <<http://dx.doi.org/10.3390/su1401.0327>>.

UNICEF. **Drone Testing for Search and Rescue Missions in Kazakhstan**. Disponível em: <<https://www.unicef.org/eca/stories/drone-testing-search-and-rescue-missions-kazakhstan>>. Acesso em: 28 de julho de 2023.

UNIVASF. **PROTOCOLO I2C**. Barramento e protocolo I2C. Brasil. 2013. Disponível em: <<http://www.univasf.edu.br/~romulo.camara/novo/wp-content/uploads/2013/11/Barramento-e-Protocolo-I2C.pdf>> Acesso em: 20 de dezembro de 2022.

VIANA, Suzana. **Um guia sobre Visão Computacional: Como os computadores enxergam?**. 2018. Disponível em: <<https://suzana-svm.medium.com/guia-visao-computacional-ae2a2ace0973>> Acesso em: 18 de dezembro de 2022.