

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
INFORMÁTICA INDUSTRIAL

FELIPE FABRICIO FISCHER DA SILVA

DETECÇÃO DE OBJETOS PARA AUXÍLIO DE ROBÔS DE SERVIÇO
HOSPITALARES

DISSERTAÇÃO

CURITIBA

2024

FELIPE FABRICIO FISCHER DA SILVA

**DETECÇÃO DE OBJETOS PARA AUXÍLIO DE ROBÔS DE
SERVIÇO HOSPITALARES**

Object Detection for the Aid of Service Robots in Hospitals

Dissertação apresentada como requisito parcial para obtenção do título de Mestre em Ciências, do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. André Schneider de Oliveira

Coorientador: Prof. Dr. Marco Antônio Simões Teixeira

CURITIBA

2024



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es).

Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



FELIPE FABRICIO FISCHER DA SILVA

DETECÇÃO DE OBJETOS PARA AUXÍLIO DE ROBÔS DE SERVIÇO HOSPITALARES

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Ciências da Universidade Tecnológica Federal do Paraná (UTFPR).
Área de concentração: Engenharia De Automação E Sistemas.

Data de aprovação: 17 de Maio de 2024

Dr. Andre Schneider De Oliveira, Doutorado - Universidade Tecnológica Federal do Paraná

Andre Roberto Ortoncelli, - Universidade Tecnológica Federal do Paraná

Dra. Vivian Cremer Kalempa, Doutorado - Fundação Universidade do Estado de Santa Catarina (Udesc)

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 17/05/2024.

Dedico este trabalho a minha família e aos meus
amigos, as pessoas incrivelmente importantes
da minha vida.

AGRADECIMENTOS

Gostaria de agradecer à todas as pessoas que fizeram a existência desse trabalho possível. Várias pessoas me acompanharem nessa jornada e não conseguirei falar de todas aqui, mas gostaria de citar especialmente os meus pais, que sempre me apoiaram, incentivaram e estiveram do meu lado, sempre me guiando nas situações mais difíceis.

Queria também agradecer aos meus amigos e família, por todo o suporte, por toda a ajuda e por todos os momentos que passamos juntos, que me deram força pra continuar.

Agradeço também aos meus orientadores, pelas dúvidas, pela ajuda nos momentos mais diferentes, por todo esse caminho que traçamos juntos.

Um agradecimento geral à todas as pessoas que fizeram esse projeto possível - nós nos encontraremos mais pelas caminhadas da vida.

Algumas pessoas chamam isso de inteligência artificial, quando a realidade é que essa tecnologia irá nos melhorar. Então em vez de inteligência artificial, acredito que aumentaremos nossa inteligência. (ROMETTY, Ginni).

RESUMO

FABRICIO, Felipe. **Detecção de Objetos para Auxílio de Robôs de Serviço Hospitalares.** 2024. 77 f. Dissertação (Mestrado em Ciências) – Universidade Tecnológica Federal do Paraná. Curitiba, 2024.

O trabalho dessa dissertação de mestrado tem o objetivo de construir um sistema de detecção de objetos simplificado usando apenas uma câmera, para um robô simulado. A motivação para construção do trabalho é buscar melhorar um sistema de navegação autônoma em hospitais, visando aumentar a segurança e eficiência do transporte de medicamentos para o leito. O sistema busca detectar objetos diversos em um ambiente hospitalar e posicionar marcadores equivalente às suas posições no mapa interno do robô, buscando criar formas de auxiliar a precisão de técnicas de navegação autônoma no futuro. Há o objetivo de futuramente usar essas técnicas para um robô real e executar a entrega de medicamentos e documentos dentro da área de um hospital, visto a falta de segurança dentro do ambiente e habilitando médicos a passarem mais tempo no leito com seus pacientes. A metodologia se dividiu na obtenção de figuras e de um modelo de processamento de imagens, na construção de um algoritmo para detecção de distância utilizando apenas uma câmera, e no posicionamento de marcadores se valendo da distância detectada e da conversão de *quaternions* para ângulos de Euler. A abordagem da obtenção de figuras é validada por meio de matrizes de confusão e gráficos mAP para a certificação da qualidade da detecção de imagens, e a distância é validada comparando os resultados alcançados pelo algoritmo com medidas reais, assim verificando a qualidade do que se foi desenvolvido. As conclusões são satisfatórias e o robô é capaz de criar marcadores pelo mapa ao reconhecer objetos pela câmera, em posições próximas da realidade, obtendo uma taxa de erro do marcador menor do que meio metro.

Palavras-chave: Mapeamento. Navegação Autônoma. Processamento de Imagens. Hospital. Marcadores.

ABSTRACT

FABRICIO, Felipe. **Object Detection for the Aid of Service Robots in Hospitals**. 2024. 77 p. Dissertation (Master's Degree in Sciences) – Universidade Tecnológica Federal do Paraná. Curitiba, 2024.

The objective of this master's dissertation is to construct a simplified object detection system, using only a camera, for a simulated robot. The motivation to make this project is to aim to improve an autonomous navigation system in hospitals, looking to increase the security and efficiency when transporting medicine to patients. The system aims to enable the robot to detect various objects in a hospital environment, positioning markers equivalent to their position in the robot's internal map, aiming to improve the precision of autonomous navigation techniques. The ultimate goal is to transfer these techniques to a real robot, capable of delivering medications and documents within the hospital premises, given the lack of security inside this environment and the possibility of making the doctors able to spend more time with the patients. The methodology employed in this study comprises the acquisition of pictures and the development of an image processing model, the creation of an algorithm for distance detection using a single camera, and the placement of markers based on the detected distance and the conversion of quaternions to Euler angles. The image processing approach is validated by confusion matrixes and mAP graphs, to certify the quality of the image detection algorithm, and the distance approach is validated by comparing the achieved results with real ones, verifying the quality of what was developed. The conclusions drawn from the study are favorable, indicating that the robot is capable of generating markers on the map by recognizing objects through the camera, with positions closely approximating reality, obtaining an error margin on the marker of less than half a meter.

Keywords: Mapping. Autonomous Navigation. Image Processing. Hospital. Markers.

LISTA DE ILUSTRAÇÕES

Figura 1 – Busca e apreensão de desvio ilegal de medicamentos no Amapá.	13
Figura 2 – Representação física das camadas de uma Rede Neural no Centro de Ciências Copérnico.	15
Figura 3 – Imagem do robô Pharmy em funcionamento pelos corredores do Habitat Senai.	31
Figura 4 – Diagrama de conexões presente no robô móvel.	32
Figura 5 – Geometria dos polos presentes dentro do estator da <i>hub motor</i>	33
Figura 6 – Campo de visão equivalente de um LiDAR.	34
Figura 7 – Mapa completo do Habitat Senai.	36
Figura 8 – Robô simulado com cinco lasers.	37
Figura 9 – <i>Frame</i> da câmera vinda do tópico correspondente no ROS, pós processamento no OpenCV.	38
Figura 10 – AMPharmy, em navegação pelos corredores do Habitat Senai.	39
Figura 11 – As principais salas do mapa, criadas dentro do software de simulação CoppeliaSim.	40
Figura 12 – A versão simulada do AMPharmy.	40
Figura 13 – Uma foto sem informações relevantes.	43
Figura 14 – Imagem com vários exemplos de <i>data augmentation</i> com o <i>dataset</i> utilizado.	44
Figura 15 – Exemplo de descongelamento de <i>layers</i> diferentes, com taxas de aprendizado diferentes, para geração de resultados diferentes.	46
Figura 16 – Objetos reconhecidos, com suas <i>bounding boxes</i> correspondentes.	47
Figura 17 – Exemplo de uma figura no grupo de testes.	49
Figura 18 – Várias imagens mostrando os <i>labels</i> reconhecidos no universo simulado. . .	50
Figura 19 – A Matriz de Confusão do modelo novo.	51
Figura 20 – Curvas representando os resultados do modelo.	52
Figura 21 – O marcador da <i>label Wheelchair</i> , visto no RViz.	57
Figura 22 – Representação gráfica dos ângulos de <i>Roll</i> , <i>Pitch</i> e <i>Yaw</i> em uma aeronave. .	68
Figura 23 – Exemplo de envio de mensagens pelo ROS por um <i>publisher</i> e um <i>subscriber</i> .	70
Figura 24 – <i>Desktop</i> do Ubuntu 20.04.	72
Figura 25 – CoppeliaSim em execução.	73
Figura 26 – A proposta de funcionamento da E-ELAN baseado em modelos mais antigos.	77

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
EPS	<i>Encapsulated PostScript</i>
PDF	Formato de Documento Portátil, do inglês <i>Portable Document Format</i>
PS	<i>PostScript</i>
UTFPR	Universidade Tecnológica Federal do Paraná
NASA	Aeronáutica Nacional e Administração Espacial, do inglês <i>National Aeronautics and Space Administration</i>
XML	Linguagem de Marcação Extensível, do inglês <i>Extensive Markup Language</i>
LTS	Suporte de Longo-Termo, do inglês <i>Long-Term Support</i>
RGB	Vermelho, Verde, Azul, do inglês <i>Red, Green, Blue</i>
SSD	Detetor de Disparo Único, do inglês <i>Single-Shot Detection</i>
ELAN	Rede de Agregação Eficiente de Camadas, do inglês <i>Efficient Layer Aggregation Network</i>
E-ELAN	Rede de Agregação Eficiente de Camadas Extendida, do inglês <i>Extended Efficient Layer Aggregation Network</i>
NR-12	Norma de Regulação 12
USB	Porta Serial Universal, do inglês <i>Universal Serial Bus</i>
API	Interface Programável da Aplicação, do inglês <i>Application Programming Interface</i>
mAP	Média da Precisão Média, do inglês <i>mean Average Precision</i>
IoU	Interseção sobre União, do inglês <i>Intersection over Union</i>
LARS	Simpósio Latino-Americano de Robótica, do inglês <i>Latin-American Robotics Symposium</i>

ACRÔNIMOS

Gimp	Programa de Manipulação de Imagem GNU, do inglês <i>GNU Image Manipulation Program</i>
ROS	Sistema Operacional Robô, do inglês <i>Robot Operational System</i>
LiDAR	Detecção de Luz e Distância, do inglês <i>Light Detection And Ranging</i>
GPL	Licença Pública Geral, do inglês <i>General Public License</i>
V-REP	Plataforma de Experimentação de Robô Virtual, do inglês <i>Virtual Robot Experimentation Platform</i>
OpenCV	Biblioteca de Visão Computacional de Código Aberto, do inglês <i>Open Source Computer Vision Library</i>
CUDA	Arquitetura Unificada de Dispositivos de Computação, do inglês <i>Compute Unified Device Architecture</i>
YOLO	Você Olha Apenas Uma Vez, do inglês <i>You Only Look Once</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVOS	16
1.1.1	Objetivos Específicos	16
1.2	JUSTIFICATIVA	16
1.3	ESTRUTURA DO TRABALHO	17
2	REVISÃO DA LITERATURA	18
2.1	ROBÔS HOSPITALARES	18
2.2	PROCESSAMENTO DE IMAGENS	21
2.3	NAVEGAÇÃO AUTÔNOMA	24
2.4	CONSIDERAÇÕES	27
3	O ROBÔ AMPHARMY	29
3.1	ABORDAGEM PROPOSTA	29
3.2	EQUIPAMENTOS	30
3.2.1	Hub Motors	30
3.2.2	LiDAR	34
3.2.3	Câmera	36
3.3	O ROBÔ REAL	37
3.4	O ROBÔ SIMULADO	38
4	PERCEPÇÃO VISUAL	42
4.1	RESULTADOS	49
4.1.1	Da Detecção de Objetos	50
5	IDENTIFICAÇÃO DE DISTÂNCIA VIA CÂMERA	54
5.1	ALGORITMO PROPOSTO PARA O CÁLCULO DE DISTÂNCIA	54
5.2	ALGORITMO PROPOSTO PARA O POSICIONAMENTO DOS MAR- CADORES	55
5.3	RESULTADOS	56
5.3.1	Do Cálculo da Distância	56
5.3.2	Posicionamento dos Marcadores	57
6	CONCLUSÕES E PERSPECTIVAS	59
	REFERÊNCIAS	60
	A – APÊNDICE	68
A.1	ÂNGULOS DE EULER E TAIT-BRYAN	68
A.2	FIRMWARE	69
A.2.1	ROS	69
A.2.2	Ubuntu 20.04	71
A.2.3	CoppeliaSim	72
A.2.4	RViz	74
A.2.5	OpenCV	74

A.2.6	Ultralytics YOLOv8	75
	GLOSSÁRIO	68

1 INTRODUÇÃO

Um ambiente hospitalar, em geral, se destaca por ter uma quantidade grande de atividades ocorrendo de maneira simultânea: uma vasta quantidade de processos, de necessidades, de entregas, interações entre o trabalhador e o paciente. Acaba por ser desafiador manter a ordem e o controle de todas as operações sendo realizadas em um cenário como esse descrito, um problema que se eleva de maneira exponencial em um ambiente medicinal, caracterizado por abrigar diversas situações emergenciais distintas, cada uma com sua necessidade, onde nenhuma delas pode ser ignorada e todas devem ser tratadas com o devido respeito.

O potencial de uso de um robô móvel dentro de um ambiente hospitalar acaba por ser muito amplo, se demonstrando ser extremamente promissor. Equipados com funções avançadas de navegação e detecção, esses robôs podem ajudar em uma gama de tarefas. Sua capacidade de operar de forma autônoma ou ser controlada remotamente pela equipe do hospital mostra elevada flexibilidade e conveniência. Robôs móveis podem reduzir o risco de contaminação cruzada, promovendo assim ambientes mais higiênicos e seguros para pacientes, visitantes e equipe médica. Com uma evolução consistente na tecnologia necessária, se espera que esses robôs tenham um papel cada vez maior na otimização dos processos hospitalares e na melhoria da qualidade das funções oferecidas.

Alguns exemplos de tais funções podem ser no uso do equipamento para simplificar tarefas mundanas, como a limpeza, manutenção de corredores ou o corte da grama, o transporte de documentos, materiais e suprimentos que outrora necessitariam um funcionário para realizar tal tarefa, a realização de reuniões de forma virtual, ou mesmo a automatização de funções a serem constantemente repetidas nesse ambiente.

Também pode-se citar a própria interação com o cliente: em um ambiente com várias empresas, um robô móvel é capaz de atuar como guia, introduzindo possíveis interessados nas várias possibilidades localizadas dentro de um ambiente de *co-working*, de uma forma interativa e bem explicativa, apresentando futuros negócios e informações. A utilização de robôs móveis em um ambiente empresarial de *co-working* pode trazer uma imagem inovadora e moderna para a empresa, o que pode ajudar a atrair novos clientes e colaboradores.

Pensando em todos esses assuntos, mas agora voltados para um ambiente dentro de um hospital, se cria a motivação desse trabalho, pensando na existência de vários problemas dentro de um sistema de trabalho dentro de um ambiente hospitalar. E o que mais se foca em um

ambiente desse porte é a farmácia, que é uma das partes mais custosas ao hospital, devido ao alto valor de manutenção e compra de novos medicamentos (WANDALKAR *et al.*, 2013).

Sendo assim, um local de tamanha importância dentro do âmbito hospitalar vem com seus problemas. Um deles, bem visível em sua estrutura, é o desaparecimento de medicamentos dentro da área da farmácia, sendo desviados e vendidos separadamente por outros fins e meios (GALVÃO, 2020). A questão de segurança é um ponto importante a ser discutido, para evitar casos como o descrito. Casos de desvio são amplamente investigados pela Polícia Federal (FEDERAL, 2023), com extensivas buscas voltadas ao combate da prática ilegal. Um carregamento de medicamentos desviados pode ser visto na Figura 1.



Figura 1 – Busca e apreensão de desvio ilegal de medicamentos no Amapá.

Fonte: (FEDERAL, 2023).

Para isso, o robô AMPharmy foi desenvolvido, pelo autor, em parceria com a startup curitibana 33Robotics, sendo esse robô o foco principal desse trabalho. Seu maior objetivo é o transporte de medicamentos de dentro da área hospitalar para os leitos dentro do hospital. Com esse transporte automatizado e facilitado, a única conexão humana com os medicamentos acaba por ser na hora da carga e descarga dos utensílios no robô, aumentando e muito a questão da segurança e tornando mais difícil os desvios de medicamentos dentro do hospital.

Para o robô conseguir fazer o transporte autônomo de medicamentos e equipamentos relacionados, ele precisa ser capaz de realizar uma navegação autônoma de confiança. Seu sistema não pode ter erros, não pode cometer falhas, ou ele colocará vidas em risco dentro do

hospital. Por mais que o sistema atual, desenvolvido com Dijkstra e um algoritmo de Monte Carlo, baseado em ROS Noetic, seja um modelo de alta confiabilidade, essa dissertação vem com um projeto que o pode elevar à um patamar superior, desenvolvendo técnicas de reconhecimento de objetos e cálculos de distância com base em visão computacional para poder auxiliar métodos de navegação autônoma a terem uma confiabilidade maior, especialmente dentro do hospital.

Com o desenvolvimento do algoritmo de detecção e cálculo de distância de imagens, esse trabalho visa como objetivo principal ser um sistema que possa auxiliar sistemas de navegação autônoma já presentes a uma plataforma superior, mais avançada e com maior precisão e confiabilidade. O trabalho, baseado num hospital, visa ser capaz de auxiliar a navegação de um robô entre diversos pontos demarcados sem qualquer interferência humana, permitindo que o trabalhador ou o médico responsável possa focar mais nos projetos e pacientes em questão que ele deve dar atenção e menos em tarefas mundanas ou repetitivas que tirariam seu tempo.

Para isso, o trabalho se vale do uso de redes neurais. Uma rede neural (ALBAWI *et al.*, 2017) é um modelo matemático inspirado no funcionamento do cérebro humano. Consiste em uma coleção interconectada de unidades computacionais chamadas "neurônios" ou "nós". Cada neurônio recebe entradas ponderadas, aplica uma função de ativação não-linear e produz uma saída.

Essas redes são organizadas em camadas. Nessas camadas, a "entrada" é responsável por receber os dados, e as camadas intermediárias (ou "camadas ocultas") são responsáveis por processar esses dados. Com isso, o sistema é capaz de aprender padrões complexos, na base do repetimento constante de padrões. A última camada, chamada de camada de saída, gera a predição final. O treinamento de uma rede neural é realizado para ser capaz de sempre gerar uma saída, quando se passa uma entrada: um exemplo clássico é a determinação de um tipo de uma flor (que será a saída do sistema) ao se passar dados característicos dela como entrada. Uma representação visual da passagem por camadas de uma rede neural pode ser vista na imagem 2, em um sistema construído fisicamente para representar de forma simples ao usuário a passagem por camadas, localizado no Centro de Ciências Copérnico, em Varsóvia, na Polônia.

Para a preparação correta da rede, são passados dados de teste e treinamento, os quais são utilizados para fazer um *matching* e verificar o nível de correspondência dos dados. Os dados de treinamento irão fazer a preparação da rede neural, executando-a e passando por suas camadas repetidamente (por sistemas classificados como "épocas"), fazendo o chamado aprendizado de máquina. Uma vez que o sistema termina de passar pelo número de épocas correspondentes, ele



Figura 2 – Representação física das camadas de uma Rede Neural no Centro de Ciências Copérnico.

Fonte: O Autor.

permite ao usuário fazer a verificação dos dados de saída obtidos com os dados de teste: assim, portanto, podendo analisar a qualidade da Rede Neural ali obtida, e seu nível de confiança.

O processo de treinamento, que corresponde à passagem pelas camadas intermediárias, envolve a apresentação repetida de dados de treinamento à rede, ajustando os pesos das conexões entre os neurônios através de algoritmos de otimização. Por cada época em que o treinamento passa, ele recebe parâmetros minimamente variáveis, com o objetivo de variar a saída da Rede e obter resultados com maior confiabilidade. O objetivo é minimizar a diferença entre as predições da rede e os valores reais associados aos dados de treinamento.

Além disso, o robô se propõe a aumentar um tempo precioso nos quais os médicos possam ficar junto dos pacientes nos leitos. Dentro de um ambiente médico, cada segundo em que o doutor possa auxiliar o paciente importa. Nesse tempo, é de suma importância que o responsável não deixe o leito para buscar remédios: nisso, o AMPharmy se propõe a fazer a

entrega de medicamentos e documentos por conta própria, aumentando o tempo do médico ao lado do paciente.

1.1 OBJETIVOS

Essa dissertação tem como objetivo realizar o desenvolvimento de um sistema que aumenta a precisão da navegação autônoma do robô, realizando o transporte de medicamentos e outros suprimentos de forma confiável.

1.1.1 Objetivos Específicos

1. Gerar um universo simulado representativo de um hospital, o qual contenha toda a área de um andar de um ambiente hospitalar e respectivos pontos de interesse, e uma versão simulada do robô AMPharmy;
2. Criar um *dataset* específico ao ambiente hospitalar correspondente ao universo simulado gerado, mostrando suas particularidades e pontos diversos;
3. Desenvolver uma maneira de se poder colocar os pontos específicos identificados em um *dataset* gerado dentro do mapa global da região do hospital, criando assim um *link* entre pontos no mapa e *labels* do *dataset*.
4. Posicionar marcadores representativos de objetos conhecidos em um ambiente simulado, podendo calcular a distância do robô ao objeto apenas usando câmeras de maneira satisfatória.

1.2 JUSTIFICATIVA

O universo da robótica é um mundo com um vasto potencial e rico leque de ambições, se expandindo rapidamente. A robótica afeta a vida de diversas pessoas, auxiliando suas necessidades, trabalhos e lazer. Todos os dias essas aplicações aumentam, globalmente (WEBSTER; IVANOV, 2020), criando uma indústria ativamente essencial para o crescimento humano e a Indústria 4.0, a qual esse inteiro universo da robótica é parte crucial (JAVOID *et al.*, 2021). A utilização de robôs e de sua implementação e essencialismo na nova revolução industrial se deve pela automatização de tarefas mundanas, realizando-as de forma autônoma de modo a aumentar

a eficiência e reduzir custos operacionais. Essa revolução acaba por aumentar a utilização da robótica nos mais diversos campos a cada dia, como na área alimentícia (IQBAL *et al.*, 2017) ou médica (YANG *et al.*, 2017), no sensoriamento de áreas distantes e perigosas, onde o ser humano tem dificuldade de acesso (ZHAO *et al.*, 2017). Também pode-se destacar coisas como a movimentação de um maquinário pesado (OHASHI *et al.*, 2016), onde, no caso, seria uma tarefa que o ser humano é completamente impossibilitado de fazer sem algum tipo de auxílio. Com a utilização de uma navegação autônoma confiável, pode-se assegurar que o robô movimente-se pelo ambiente ao fazer essas tarefas de forma segura e eficiente.

A justificativa para a realização deste trabalho se dá pela construção de um algoritmo que seja capaz de complementar a navegação autônoma de um robô usado em ambiente hospitalar, visando, com esse robô, aumentar a segurança e a confiabilidade da entrega de medicamentos para o leito. O trabalho usa um algoritmo combinando uma navegação autônoma, dada pelas bibliotecas do ROS (*Robot Operational System*) em conjunto com sensores Hall e um LiDAR, com alternativas voltadas ao processamento de imagens, se valendo de bibliotecas Python e câmeras bem posicionadas, além de todas as etapas de validação que envolvem um ambiente simulado. Essa dissertação busca a implementação do sistema finalizado em operação em um ambiente hospitalar num universo experimental, e de sua validação baseado na análise de resultados conseguidos posteriormente.

1.3 ESTRUTURA DO TRABALHO

Esse texto está organizado em diversos capítulos, nos quais se tenta abranger todos os tópicos relevantes para o entendimento claro do que é o projeto e como ele é feito e executado.

O capítulo 1 está fundamentado em ser uma introdução sucinta ao que é o projeto. No capítulo 2 se apresenta toda a fundamentação teórica por trás do trabalho, com uma revisão da literatura e do estado da arte. Já no capítulo 3, se apresentam conceitos gerais, mostrando um pouco da base de onde se inspirou o projeto, tem-se a apresentação do funcionamento do projeto em si, além dos equipamentos diversos usados. No capítulo 4, o trabalho disserta à fundo sobre o modelo construído para a detecção de imagens, enquanto o capítulo 5 mostra uma explicação sobre o modelo construído para a detecção de distâncias com processamento de imagens e posicionamento de marcadores. O capítulo 6, por fim, termina o trabalho em uma conclusão. Há também um apêndice entrando em detalhe sobre os materiais e programas utilizados em seu planejamento.

2 REVISÃO DA LITERATURA

Essa seção terá um destaque para a descrição do atual estado da arte dos conceitos utilizados dentro dessa dissertação, buscando fazer comparativos com as atuais referências e modelos para as áreas mencionadas, mostrando as inovações deste projeto com base no que já existe publicado.

2.1 ROBÔS HOSPITALARES

Sobre a literatura envolvendo robôs hospitalares, existe uma ampla variedade de trabalhos que podem ser observados e analisados. O trabalho em (EVANS, 1994), por mais que seja mais antigo, é um caso de estudo relevante para o manuscrito atual, onde pode-se até ver um projeto mais antigo para se fazer uma comparação da evolução da tecnologia para o que se há hoje em dia. Nele, pode-se ver o desenvolvimento de um robô de suporte de entrega de medicamentos e objetos pelo hospital, um conceito semelhante ao que essa dissertação se propõe. Porém, o projeto do HelpMate possui algumas restrições que não são vistas no AMPharmy, como o uso de sinais de radiofrequência de espalhamento espectral para a comunicação (TORRIERI, 2005), além de um uso de radiofrequência ao invés de um sensor infravermelho para a detecção de objetos.

No trabalho de (KYRARINI *et al.*, 2021), é possível visualizar uma evolução em robôs de cuidados pessoais, assistivos e de reabilitação, envolvendo toda a gama de robôs hospitalares. Há uma variedade diversa de robôs hospitalares mencionados nesse artigo, onde alguns são já presentes no mercado, alguns experimentais e outros voltados à pesquisa.

Um ponto importante levantado pelo artigo é a conclusão que os autores chegam sobre a tecnologia ainda não ter atingido uma maturidade grande o suficiente. Não apenas isso, mas o desafio de levar a tecnologia para dentro de um hospital envolve outros problemas, como a privacidade dos pacientes; o risco de mal-funcionamento, performance e segurança; a aceitação de um robô substituindo uma tarefa humana pelo usuário, entre outros fatores. Acaba por ser um fator menos complicante para o projeto em questão apresentado nessa dissertação, visto que a ideia não é substituir uma tarefa executada por um humano, mas a criação de um robô que opere auxiliando o humano e tornando o trabalho hospitalar mais eficiente.

Ainda na área de robôs móveis dentro de um hospital, o artigo de (SCHAHAWI *et al.*,

2021) faz uma análise semelhante, mas voltada especificamente à relevância de robôs móveis dentro da área hospitalar fazendo a limpeza de ambientes contra vírus e bactérias com o uso de luzes ultra-violeta, possuindo um pico de relevância na época da pandemia da COVID-19. Um problema detectado, não apenas com essa tarefa, mas com qualquer tipo de robô móvel dentro de um hospital, é a falta de preparo dos locais em relação à *design* e inventário, podendo causar problemas para a navegação, além da criação de obstáculos diversos causados pelo tráfego e permanecimento de pacientes e funcionários pelas alas do hospital. Por mais que seja um problema de grande relevância, é algo que é minimizado pela aplicação do LiDAR no trabalho aqui descrito, onde, por ser um LiDAR de alta qualidade, sendo ele o SZ-16D da empresa japonesa Keyence, utilizado na versão física do AMPharmy, o equipamento é capaz de detectar obstáculos e rotas difíceis com facilidade, reprogramando-as conforme necessário. O SZ-16D tem características marcantes como alta precisão do mapeamento e uma distância do laser que vai até quarenta metros.

A pesquisa descrita em (HOLLAND *et al.*, 2021) é um ponto de partida mais voltado à robôs de serviço dentro do setor de auxílio hospitalar e uma visão geral de tecnologias relacionadas, trazendo exemplos de robôs móveis voltados à limpeza, esterilização, comunicação remota, entre outros fatores. Sobre especificamente a parte de logística hospitalar com robôs, o trabalho traz a preocupação relevante sobre a substituição de ofertas de emprego por seres humanos, o que não deve ser um risco para o desenvolvimento do AMPharmy quando se considera que o intuito do projeto não é a substituição de pessoas em seus empregos, mas a criação de aparelhos que possam auxiliá-las em suas tarefas. Um ponto de observação que deve ser considerado é a otimização de tarefas dentro da área hospitalar, levando a um aumento de produtividade e eficiência no ambiente.

Há um detalhamento similar em (MARMAGLIO *et al.*, 2023). O trabalho em questão faz também uma análise de vários robôs autônomos de serviço dentro do âmbito hospitalar, mas também se voltando para a análise de drones no mesmo cenário. Esse artigo faz menções detalhadas de vários aspectos técnicos desses robôs, como a presença de sensores LiDAR, câmeras, sensores de temperatura, umidade, acelerômetro, giroscópio e GPS, além de preocupações com a vibração do robô. Equipamentos como acelerômetro, giroscópio não foram considerados para esse projeto, pois acabavam saindo do escopo do que foi proposto.

Algo similar pode ser visto em (SHIEH *et al.*, 2004), onde o trabalho em questão possui uma proposta parecida, da construção de um robô hospitalar de serviço. Esse também é um

projeto que faz do uso de sensores ultrassônicos para a detecção de obstáculos pela região geral do robô. Pode-se também falar sobre o controle de navegação: enquanto o projeto descrito nessa dissertação se vale de um algoritmo adaptado de Monte-Carlo, o projeto descrito desenvolveu uma estratégia própria de navegação se valendo de controles *fuzzy*.

Pode-se ver em (HUNG *et al.*, 2021) uma estratégia um pouco diferente, porém. Ao invés de buscar auxiliar os trabalhadores do hospital com a entrega de medicamentos e utensílios, o robô PARO entra na esfera social, tentando auxiliar pacientes com demência em uma escala mais terapêutica. Porém, um empecilho considerável do PARO é que ele não foi testado em uma escala real, apenas com voluntários e grupos específicos, e que os resultados da interação humana são inconclusivos para decidir se o robô é peça fundamental na recuperação ou não. Isso se aplica em uma vantagem do AMPharmy, que não interage de forma humana com pacientes ou funcionários: a função do robô se volta a ser meramente utilitária, e não busca substituir ou auxiliar em questões envolvendo emoções ou similares.

Uma vista no que é discutido em (CARDONA *et al.*, 2020) pode mostrar um estudo mais à fundo sobre robôs móveis dentro de hospitais, entrando, novamente, do conceito da pandemia da COVID-19, mas expandindo o conceito para aplicações mais abrangentes. São discutidas opções sobre telepresença, limpeza, desinfecção e entrega de equipamentos dentro de hospitais. A construção desse projeto entra na discussão sobre a entrega de equipamentos, onde o texto em questão desenvolve uma argumentação sobre como esse tipo de robô pode ser útil também em um cenário onde a pandemia possa tomar uma escala maior, e o contato entre o paciente e o médico se torne cada vez mais perigoso.

No contexto do trabalho de (TAYLOR *et al.*, 2019), tem-se uma detalhada explicação sobre a utilização de robôs móveis para o auxílio de enfermeiras na linha de frente do hospital, com um robô equipado com suporte para ajudar na respiração de pacientes com dificuldades e outros equipamentos cirúrgicos. São levantadas dificuldades sobre a integração do robô no conceito humano e problemas relacionados à privacidade do paciente, mas também se levanta um ponto importante sobre a precisão do trabalho em equipe para fazer a execução do modelo funcionar de forma eficiente. Quanto à isso, o AMPharmy também é dependente desse quesito: uma equipe que não esteja trabalhando em conjunto não conseguirá extrair o potencial máximo do robô e acabará esbarrando em problemas de coordenação com o equipamento.

O artigo descrito em (NAVYA; RANJITH, 2021) entra explicitamente na discussão de *path planning* dentro de um hospital, especialmente para robôs de serviço. Ao invés do

Dijkstra, porém, utilizado nesse trabalho na versão física do AMPharmy, o artigo em questão se baseia em modelos de *Depth-First Search* e *Breadth-First Search*, respectivamente recursivos e não-recursivos. Porém, a utilização desses algoritmos acaba sendo pouco eficiente, consumindo uma parcela eficiente de memória e tempo em comparação ao algoritmo utilizado na versão física do AMPharmy, especialmente porque o algoritmo Dijkstra tende a ser mais capaz de encontrar rotas curtas em tempo menor devido ao seu cálculo ponderado.

Uma estratégia interessante sobre robôs móveis e a logística de hospital pode ser lido no trabalho em (LAPPALAINEN, 2019). Na questão do trabalho envolvido nesse cenário, o robô não foi construído pelos organizadores do projeto, mas adaptado de um outro projeto: o modelo TUG, desenvolvido pela empresa *American Aethon*, e re-desenvolvido para trabalhar com logística hospitalar. As conclusões de uso são que o robô teve um impacto causado maior em algumas áreas, nas quais se incluem o transporte de objetos variados pelo hospital, estando de acordo com o propósito do projeto aqui desenvolvido.

2.2 PROCESSAMENTO DE IMAGENS

Já mostrando os avanços na detecção de métodos com imagens, pode-se avaliar uma outra gama de trabalhos relacionados que mostram um olhar mais atento à esse ramo. Por exemplo, há o trabalho de (ALKENDI *et al.*, 2021), que mostra exemplos de estado da arte de odometria visual, fazendo uma análise mais a fundo deles e do funcionamento dedicado de cada uma delas. O artigo em questão entra em detalhes em vários modelos diferentes, principalmente VO (Visual Odometry) e VIO (Visual-Inertial Odometry).

Algumas conclusões importantes que o artigo chega são sobre a importância dessas técnicas para determinar o posicionamento de objetos em um plano espacial determinado. Nessas técnicas, se é mencionado equipamentos utilizados nessa dissertação, como câmeras e LiDARs, mas também outras opções que não foram utilizadas devido ao escopo do projeto, mas que podem ser adicionadas em versões futuras, como câmeras térmicas, IMUs e radares.

O artigo em (SALVI *et al.*, 2021), sendo analisado em mais detalhes, traz uma discussão relevante sobre análises de imagens digitais, com o uso de *deep learning*, antes e após serem processadas, com o uso de CNNs (Redes Neurais Convolucionais). Quanto ao pré-processamento, o artigo cita sobre uma curadoria dos dados, que tende a ser extremamente relevante, fazendo uma preparação dos dados a serem usados no modelo de modo a ter resultados mais robustos, o que também é feito no modelo utilizado nessa dissertação. São citados métodos para normalização

de cores, detecção de curvas e técnicas para seleção de caminhos.

Em (SHIT, 2020) é possível ver algo parecido, onde metodologias de estado da arte de navegação autônoma são verificadas em comparação em uma com a outra, analisando suas limitações e capacidades de evolução e utilização futura. São exemplos visíveis da constante evolução das técnicas de processamento digital de imagens. Com esses artigos, é possível visualizar que o uso da visão computacional é capaz de ser importante para vários campos, incluindo áreas relacionadas à robótica hospitalar: esses conceitos conseguem ser aplicados no projeto em questão. Na parte relevante ao pós-processamento, o artigo fala sobre métodos de classificação para redes de *Deep Learning*, citando também a importância desses métodos para gerar resultados com alta precisão em *Transfer Learning*. O texto cita que a técnica de *Transfer Learning* tem uma precisão acima do elevado, e, portanto, é utilizada em quase todos os exemplos providenciados. Também há citações sobre *data augmentation* na parte de pós-processamento para gerar resultados altos de precisão no final. Todos esses métodos também são aplicados no desenvolvimento dessa dissertação.

Em (XIAO *et al.*, 2020), pode-se ver uma pesquisa sobre a remoção do ruído causado por objetos não-desejados no fundo de uma imagem detectada, chegando numa conclusão de que um fundo pode afetar a detecção de uma imagem em quase noventa por cento dos casos, mas, que, por outro lado, esse efeito pode não ser prejudicial: o artigo não chega em uma conclusão definida, mas concorda que os fundos podem muitas vezes afetar de maneira positiva a detecção de uma imagem. Nesse trabalho, como o *dataset* foi gerado a partir de um ambiente simulado, os fundos acabam por ser simples, e, portanto, menos ruído é causado para a detecção de objetos em si, causando menos problemas na hora de reconhecer uma imagem.

Já o trabalho de (NGUGI *et al.*, 2021) entra em detalhes maiores sobre processamento de imagens, com uma análise profunda em técnicas se valendo desses modelos para o controle de pestes em folhas na agricultura. O artigo discute também problemas alcançados quando desenvolvendo o trabalho. Dentre eles, podem-se destacar a falta de imagens para o *dataset*, anomalias na detecção de sintomas visualmente parecidos e reconhecimento de problemas em partes diferentes da imagem. São problemas similares aos encontrados nessa dissertação. O artigo em questão faz sugestões para resolver esse problema, como *Data Augmentation* e a segmentação das imagens, que também são usadas nesse projeto.

O projeto visto em (JOSHI *et al.*, 2020) mostra um uso de reconhecimento de imagens para pessoas com dificuldades visuais. Nele, os autores se valem de uma placa Jetson Nano

e uma câmera de Raspberry Pi. O modelo foi treinado com base em uma CNN e milhares de imagens retiradas de serviços de busca *online*. O método de geração do modelo é similar ao usado nessa dissertação, treinado com base em um modelo YOLO, mostrando a eficiência do método escolhido.

Já no artigo visto em (ZAMAN *et al.*, 2019), o que se pode ver é a execução de odometria visual para a área da agricultura, sendo capaz de visualizar e monitorar a plantação com boa eficiência computacional. Nesse caso, a odometria visual é dada por um sistema NCC (*Normalized Cross-Correlation*), adaptado para a movimentação da máquina, testado em vários tipos de terreno diferentes. O sistema se provou ter uma acurácia boa nas situações verificadas, mostrando que a navegação com informações vindas de um processamento de imagem é possível e viável.

Também na área relacionada à plantio e agricultura, o trabalho em (TANG *et al.*, 2020) cita um caso onde métodos de processamento de imagens são utilizados para a coleta de frutas. Nesse caso, métodos de RGB-D são colocados em uso pelos autores, como no caso do Kinect, além dele ser utilizado em conjunto com lasers, câmeras multiespectrais e outros sensores. A câmera é controlada por um braço robótico, colocado na traseira de uma caminhonete que passa pelas plantações, capaz de coletar as frutas ao reconhecê-las, usando um método de *deep learning* onde o sistema recebe o *feedback* de seus erros e melhora seu próprio sistema baseado nisso, reconstruindo a região detectada em um *software* de simulação 3D. O sistema teve problemas identificados principalmente com a iluminação do ambiente alterando a percepção da câmera, o que é um problema que tende a ser nulo no ambiente simulado e reduzido em ambiente *indoors*, que é o ambiente de operação do AMPharmy. Também é citado uma dificuldade de reconhecer frutas maduras e o problema já mencionado de um número reduzido de *samples* para a detecção de objetos. Esses textos são necessários para se entender exemplos de dificuldades que podem ser encontrados em sistemas de processamento de imagens, especialmente relacionados à baixa disponibilidade de *samples*.

Observando o trabalho de (GUAN; LIU, 2021), pode-se analisar técnicas de processamento e análise de imagens com uso de *machine learning* para um ambiente hospitalar. Nele, tenta-se valer de CNNs (Redes Neurais Convolucionais) e DANNs (Redes Neurais Adversariais de Domínio) para fazer a divisão de dados relevante ao modelo do artigo. Uma discussão relevante desse artigo para a dissertação atual é sobre a construção de um modelo com número limitado de imagens, onde se conclui que isso acaba por trazer um grande desafio para um

treinamento adequado de modelos. Nessa dissertação, ideias de modelos como *Transfer Learning* tentam contornar isso de uma forma com que a acurácia do modelo se mantenha alta.

O artigo de (BUDD *et al.*, 2021) também envolve o processamento de imagens com a área medicinal. Nele, há um estudo sobre técnicas de *Machine Learning* e *Active Learning*, e um comparativo entre os dois modelos para *Deep Learning*. Ambos tem um bom funcionamento, mas requerem um humano para um resultado positivo. Ainda assim, o artigo conclui que a utilização de *Deep Learning* para técnicas de processamento de imagem no ramo hospitalar pode gerenciar uma mudança completa no panorama de como tarefas clínicas são feitas, o que entra de acordo com o desenvolvimento desse projeto e com o uso de *Deep Learning* para a identificação de imagens no hospital.

Em (BHATTACHARYA *et al.*, 2021), há uma abordagem preparada para a pandemia da COVID-19 dentro da área hospitalar, se valendo de técnicas de treinamento supervisionado, não-supervisionado e com *Reinforcement Learning*. O objetivo dos autores é fazer um desenvolvimento de um processamento de imagens voltado à identificação de sintomas para a identificação da doença, além de tentativas de predição para expansões da pandemia e possibilidade de rastreamento da doença com imagens. Um problema que os autores encontraram com o trabalho foi, também, a presença de um *dataset* limitado.

Analisando o trabalho de (MELOTTI *et al.*, 2020), também pode-se ver uma construção de ideias para o reconhecimento de imagens usando um conjunto de câmera com um LiDAR, similar ao que esse trabalho se propõe. Uma estratégia proposta nesse trabalho é a utilização de uma visualização 3D, por meio de LiDARes de *point clouds* ou *depth maps*, se valendo de profundidade. A distância real dos objetos, assim como nessa dissertação, é dada pelo valor lido pelo LiDAR. Os melhores valores encontrados pelo projeto foram os encontrados com uma câmera de profundidade. Uma câmera de profundidade acaba ficando fora do escopo do projeto dessa dissertação, mas é algo a ser amplamente considerado quando se pensar em versões futuras do robô.

2.3 NAVEGAÇÃO AUTÔNOMA

A base para se ter um robô móvel com funcionamento adequado para técnicas de navegação autônoma (ALKENDI *et al.*, 2021), para se localizar um robô móvel em qualquer ambiente, é uma odometria adequada, um mapeamento eficiente do ambiente referente, uma configuração correta dos motores para seu controle, um planejamento eficiente das rotas em

questão para sua movimentação.

O artigo em (ARKIN; MURPHY, 1990) mostra um estudo feito há mais de trinta anos que contém uma boa relevância para a evolução no período. Nele, é possível ler sobre o desenvolvimento de um AGV (*Automated Guided Vehicle*) navegando automaticamente em um ambiente, tentando gerar um planejamento de navegação e uma rotina para estacionar o robô. O artigo usa um modelo A* para o planejamento de rotas e desvio de obstáculos, e já ali era capaz de reprogramar sua rota para o desvio de obstáculos, detectados por um sensor ultrassônico. Apesar do AMPharmy usar Djikstra, ele também é capaz de desviar de obstáculos com maestria, graças ao uso da tecnologia mais moderna combinado com um LiDAR de excelência.

Em (ZOU *et al.*, 2021), pode-se ler sobre um comparativo de técnicas de mapeamento e localização para navegação em ambientes internos, com sensores LiDAR diferentes. No caso desse artigo, várias técnicas são analisadas em detalhes, testando a validação do método de localização pela odometria da roda sozinha, AMCL, se valendo de Gmapping, Cartographer, IMU e LOAM.

A conclusão dos autores é que, por conta própria, a odometria da roda é insuficiente e não-confiável para um robô obter a própria localização, devido ao fato de ser muito sujeita à erros. Porém, quando combinada com outros fatores, como uma IMU ou um LiDAR, tende a ser mais confiável, corrigindo erros inevitáveis que surgem da roda sozinha. É a abordagem usada nesse trabalho, também, onde a odometria da roda é utilizada, porém, em conjunto com a leitura do LiDAR SZ-16D. A precisão de um modelo como o AMCL também acaba ser reduzida quando não há a presença de um mapa, mas, quando utilizada em parceria com o Gmapping, como nesse projeto, ou o Cartographer, fornecendo mapas, sua eficiência acaba por aumentar de forma exponencial.

Em (LI; IBANEZ-GUZMAN, 2020), há uma pesquisa detalhada sobre a utilização de LiDARes, especialmente voltado à navegação de veículos autônomos. O artigo chega em conclusões sobre dificuldades que projetos usando LiDARes podem chegar, mais em específico, o seu preço, padrões de segurança, funcionamento em longas distâncias e situações climáticas, resolução da imagem e tamanho. São problemas muito pertinentes, mas não tão presentes no AMPharmy: o robô não se preocupa tanto com situações climáticas ou funcionamento em longas distâncias devido ao funcionamento *indoor*. As questões de segurança foram verificadas e adequadas, especialmente se valendo do ambiente simulado para testes, e a resolução da imagem não é um problema com o LiDAR laser. O tamanho e o preço também foram ajustados para o

projeto ser comercialmente viável.

O trabalho de (DAZA *et al.*, 2020) cita uma análise mais detalhada dos sensores LiDAR usados em navegação autônoma, assim como esse trabalho utiliza, e de sistemas de localização baseados neles. Foram feitos testes com sistemas variáveis entre 3 e 6 graus de liberdade de movimento no LiDAR, permitindo-o fazer correções necessárias dos erros na nuvem de pontos. O trabalho em si é inteiramente focado na correção de erros do LiDAR e acaba por ter resultados exemplares. Porém, como dito pelos próprios autores, a aplicação de um LiDAR 3D acaba por ter resultados similares à um LiDAR móvel, e isso é uma aplicação simulada no ambiente virtual por uma detecção de obstáculos feita por cinco feixes de laser.

Algo parecido pode ser visto em (YEE *et al.*, 2020), onde o projeto envolve uma faixa de laser inclinada para possibilitar a localização de objetos de tamanho menor nos arredores do robô, que não seriam normalmente detectados. O mapeamento nesse cenário é dado também pelo *software* Gmapping, o desvio de objetos também é dado por DWA, e o algoritmo de busca e *path planning* é dado por A*. No caso, o LiDAR em referência no artigo é um RPLidar A1, capaz de detecção em 360 graus. O artigo comprova que o mapeamento acaba por ficar mais preciso se valendo da inclinação do LiDAR. Uma estratégia similar é utilizada nessa dissertação, também por meio dos cinco feixes de laser na vertical, os quais conseguem registrar faixas diferentes de detecção em alturas diversas, registrando obstáculos que poderiam ser perdidos de outra forma.

Exemplos de robôs móveis operacionais e que podem ser considerados um sucesso entram no caso de (BURGER *et al.*, 2020), onde o projeto é realizado na função do robô móvel encontrar fotocatalistas na produção de hidrogênio da água. Nele, pode-se ver uma função avançada de uso de um robô móvel operacional com navegação autônoma, com um funcionamento similar em questão de planejamento de rotas do AMPharmy.

Já em (JIANG *et al.*, 2022), tem-se também um projeto de robô móvel em uma estufa de plantas, onde a navegação do equipamento se dá por LiDARes 2D e 3D. No caso, o robô em questão usa Cartographer para a parte de SLAM, ao invés de Gmapping, como nesse projeto, e um LiDAR completamente 3D, com os dados sendo lidos em nuvem de pontos e convertidos para uma leitura 2D em LaserScan. Porém, devido às condições específicas do projeto, ele funciona apenas no ambiente especificado, tendo que ter seus parâmetros modificados até mesmo com uma mudança na velocidade do robô, o que não é algo que acontece com o AMPharmy.

Em (SUN *et al.*, 2023), há um trabalho similar, onde se propõe uma navegação de robôs móveis utilizando um LiDAR 2D. Nesse caso, o projeto se utiliza de uma Jetson Nano,

que acaba por ser mais potente do que a Raspberry utilizada na versão física do AMPharmy, além da presença de uma IMU para melhorar o posicionamento do robô. O projeto propõe técnicas avançadas e melhoradas de SLAM, chamadas de LRBPF-SLAM e GBI-RRT para o *path planning*. O projeto demonstra resultados positivos, melhores que os utilizados tradicionalmente na robótica.

Também pode-se citar o trabalho de (GUERREIRO *et al.*, 2019), onde se consegue ver um projeto com execução real, fazendo a navegação autônoma de um robô ser auxiliar para a movimentação de pessoas com deficiência visual, conduzindo um estudo com pessoas reais gerando *feedback* para o robô. No caso, o robô CaBot teve diversos testes realizados com seres humanos, os quais retornaram um *feedback* completamente positivo para a existência de um robô auxiliar convivendo pelo local: algo que acaba por ser muito bom para o projeto planejado nessa dissertação.

Em (BELLEMARE *et al.*, 2020), pode-se ver a estrutura de uma navegação autônoma confiável, utilizando balões atmosféricos e técnicas de *reinforcement learning*, integrando uma miríade de parâmetros para se ter um algoritmo confiável ao fim, sendo um exemplo também de robô de sucesso que é capaz de navegar de forma confiável nas condições mais adversas.

O trabalho de (JOO *et al.*, 2020) acaba por ser bem interessante por ter um objetivo similar ao dessa dissertação, buscando criar uma navegação autônoma se baseando em ambientes de modelação, pelo uso de pontos de referência em uma câmera, usando táticas de *reinforcement learning* e re-treinando o dataset da YOLOv3 para com as imagens do ambiente, recriando eles em um ambiente simulado e experimental. Uma vantagem que esse projeto tem sobre o artigo é a utilização de uma Ultralytics YOLOv8, ao invés da YOLOv3, que nesse ponto já era mais ultrapassada. A experiência dos autores no ambiente simulado foi bem-sucedida, provando que é possível fazer a passagem para um ambiente real no futuro.

2.4 CONSIDERAÇÕES

Após ler todos os trabalhos mencionados, várias considerações podem ser feitas, em comparação com os projetos lidos e analisados anteriormente, que também serão divididas nas seções de robôs hospitalares, processamento de imagens e navegação autônoma.

Vários artigos mencionam a questão do robô ter dificuldades com obstáculos diversos e do ambiente hospitalar em questão, onde ele teria uma dificuldade elevada em navegar. Porém, visto a utilização de cinco sensores LiDAR no robô simulado, além do auxílio da câmera, o

autor acredita que esse tipo de problema seria pouco relevante para o robô. Além disso, muito se discute sobre a possibilidade desses robôs acabarem substituindo os trabalhadores no hospital. Porém, essa é uma possibilidade difícil de acontecer, visto que a criação do AMPharmy é voltada para o auxílio de tarefas já existentes, visando o trabalhador executá-las de forma mais eficiente e que traga menos risco ao paciente.

Também pode ser lidas várias considerações sobre a impossibilidade de obter várias imagens na construção do *dataset*. No caso do AMPharmy, não apenas a versão simulada dele requer menos imagens para a construção do *dataset*, mas a utilização de *Transfer Learning* também possibilita a construção de um *dataset* muito preciso até mesmo com o robô físico, se valendo de um número menor do que o normal de imagens para a construção do *dataset*.

Outras considerações que podem ser lidas são marcadas na utilização dos projetos apresentados nos artigos referenciados, como a utilização de vários deles em A* para a construção do *path planning*, enquanto o projeto dessa dissertação enfatiza na utilização do algoritmo Dijkstra, uma vantagem especialmente relativa à otimização do caminho descoberto, consideravelmente maior quando o ambiente não possui uma heurística adequada. Além disso, a utilização de cinco sensores LiDAR no robô simulado para criar um aumento na eficiência da navegação autônoma, o que não é visto em outros trabalhos. Por fim, a utilização das técnicas nessa dissertação descritas de navegação visual são essenciais para ser a diferença, criando uma navegação autônoma ainda mais eficiente que as descritas.

3 O ROBÔ AMPHARMY

Nesse capítulo, o autor apresenta conceitos que serão utilizados ao decorrer desse estudo, considerando equipamentos utilizados e bases teóricas para a construção do projeto, além de uma introdução à proposta dessa dissertação.

3.1 ABORDAGEM PROPOSTA

Para a construção desse projeto, buscando fazer a detecção de objetos em um ambiente hospitalar, duas etapas foram consolidadas. Isso foi feito com o objetivo de fazer o projeto fluir de uma maneira mais natural, onde seu desenvolvimento pôde seguir uma ordem clara de fatores para ser concluído.

A primeira etapa se refere à percepção de ambiente por visão computacional. Nessa etapa, foi-se desenvolvido um sistema, se valendo dos métodos de *Transfer Learning*, para fazer a detecção de objetos específicos criados dentro de um *dataset*, com a versão física do AMPharmy.

Para a construção desse *dataset* e a validação do sistema, se usou de um ambiente real de um *co-working* com várias empresas, onde o método de percepção da área pôde ser capaz de fazer o reconhecimento de vários objetos e cenários específicos à esse universo, certificando que o modelo era capaz de realizar o *Transfer Learning* com sucesso e desenvolver um *dataframe* que conseguisse fazer uma percepção de qualidade com poucas imagens.

Na segunda etapa, a mesma metodologia foi utilizada para se construir um *dataset* mais voltado ao projeto em questão: isso é, um *dataset* em um ambiente hospitalar, construído virtualmente, voltado ao desenvolvimento dessa tecnologia e de seus testes. Uma representação virtual do AMPharmy foi utilizada nesse cenário, equipada com cinco sensores LiDAR.

Para isso, a versão simulada do AMPharmy foi construída não apenas com o intuito de testar a percepção de ambientes em um cenário hospitalar, mas também de desenvolver o método de identificação de distância via câmera e posicionamento de marcadores virtuais no mapa interno do robô. Com isso, tudo o que se tinha a intenção de ser desenvolvido nessa dissertação pôde ser testado no ambiente virtual.

Nesse capítulo, também será discutido a metodologia utilizada para essas abordagens, verificando equipamentos, *hardware* e conceitos utilizados para desenvolver tanto o robô virtual quanto o robô real, e o que se foi utilizado para conseguir chegar nos resultados obtidos ao final

desse projeto.

3.2 EQUIPAMENTOS

Para a execução do projeto, se utilizou como base um robô móvel, o AMPharmy, construído pela *startup* 33Robotics. O projeto dessa dissertação pôde tomar forma em parceria com a empresa, também localizada dentro do Habitat Senai. O Pharmy foi construído com o propósito de ser um robô com navegação autônoma, capaz de realizar a entrega de medicamentos e objetos diversos dentro de hospitais e leitos clínicos. O robô pode ser visto com mais clareza na imagem da Figura 3.

A versão final do Pharmy possui vários componentes importantes para sua navegação autônoma completa. Dentre esses, pode-se destacar a existência do LiDAR e dos sensores Hall presentes nas *hub motors* do robô, além da câmera USB, responsável pela detecção de imagens em tempo real para auxílio da navegação autônoma. Esses componentes, funcionando de forma conjunta, são os responsáveis pela odometria do robô, permitindo seu funcionamento de maneira exemplar, e podem ser visualizados no diagrama visto na Figura 4.

O projeto dessa dissertação, porém, envolve principalmente uma versão simulada do robô AMPharmy. A motivação para o desenvolvimento e utilização de uma versão simulada se vem por vários fatores, dentre eles incluso a possibilidade de realizar testes em um ambiente controlado. Também pode-se citar a validação de métodos de programação e controle do projeto antes de ser colocado em uma etapa real, pois testar métodos na própria máquina real sem uma pré-validação pode-se provar perigoso para com os equipamentos no robô localizados, além de custoso, pois uma bateria de testes simulada acaba por ser consideravelmente mais rápida que uma bateria de testes no projeto físico.

Nessa seção, os equipamentos necessários para as versões física e simulada do AMPharmy serão discutidas em mais detalhes, além de uma explicação necessária dos conceitos de programação, físicos e matemáticos, que regram o funcionamento desse projeto.

3.2.1 Hub Motors

Hub motors são comumente vistos em rodas de *hoverboards*, bicicletas e patinetes elétricos, métodos de transporte que se popularizaram no mercado pela metade da década de 2010 (GRUBER *et al.*, 2011), mas que possuíam estudos sobre sua efetividade desde o começo do



Figura 3 – Imagem do robô Pharmy em funcionamento pelos corredores do Habitat Senai.
Fonte: Autor.

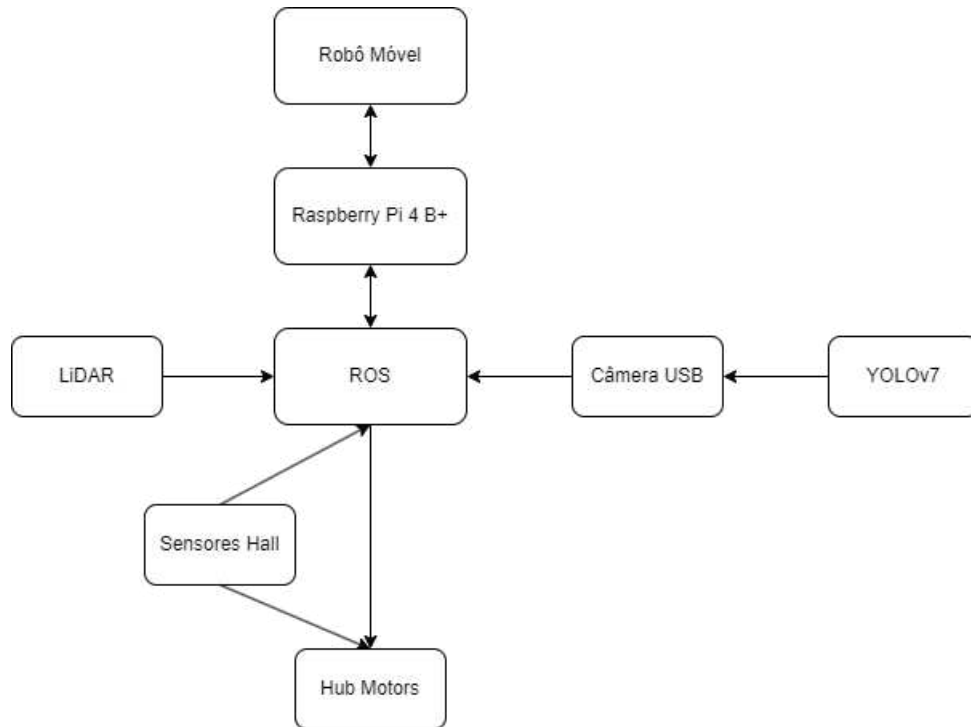


Figura 4 – Diagrama de conexões presente no robô móvel.

Fonte: Autor.

século (SCHALKWYK; KAMPER, 2006). Suas características marcantes são, principalmente, a integração do motor responsável pelo controle do movimento da roda já presente internamente dentro do equipamento, além de um tamanho e uma massa um pouco mais elevada comparada à outros motores, capaz de realizar o transporte de cargas mais pesadas.

O funcionamento de uma *hub motor* se dá pelos polos magnéticos existentes dentro da roda, com seu estator e rotor já presentes dentro de si própria. A geração do sistema que será suficiente para gerar a rotação do motor interno, e, subsequentemente, da roda, se dá pela conexão desses polos magnéticos com as fases do motor, localizados dentro do estator. O número de polos dentro da roda pode variar conforme o modelo, como pode ser visto na Figura 5, mostrando a geometria dos polos localizados dentro do estator.

A maior parte dos modelos de *hub motor* já é comercializada com a presença do sensor *hall* interna dentro de si. Em sua essência, sensores *hall* são sensores puramente magnéticos, comercializados e utilizados por eras dentro da eletrônica: já em 1995, mais de um bilhão de sensores *hall* haviam sido vendidos mundialmente (KUZE; SHIBASAKI, 1997).

Seu funcionamento é relativamente simples. Como o nome já indica, sua execução é baseada no efeito *hall* (KARPLUS; LUTTINGER, 1954), já bem documentado há quase setenta anos. Devido ao efeito da Lei de Faraday-Lenz, percebe-se que, graças à presença de um campo e fluxo magnético em determinado local, é também possível detectar uma força eletromotriz

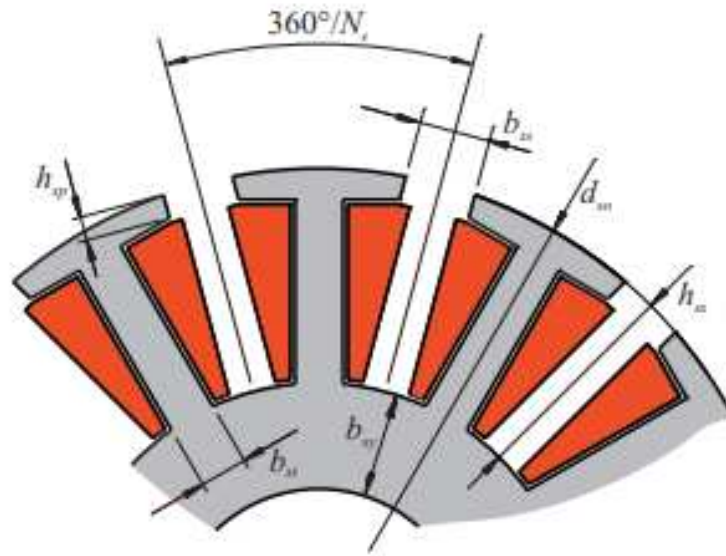


Figura 5 – Geometria dos polos presentes dentro do estator da *hub motor*.

Fonte: Wolfgang Gruber (GRUBER *et al.*, 2011).

equivalente, medida em volts, e, portanto, de corrente elétrica resultante. A equação da Lei de Faraday-Lenz pode ser vista na Equação 1, onde ϵ é o equivalente da força eletromotriz e ϕ é o delta do fluxo eletromagnético, dividido pelo delta do tempo.

$$\epsilon = \frac{\Delta\phi}{\Delta t} \quad (1)$$

Se valendo disso, os sensores *hall* são capaz de detectar variações eletromagnéticas dentro de seu circuito (CALKINS *et al.*, 2007). Com a utilização de um ou mais ímãs dentro da roda, em rotação com o motor, sensores *hall* são colocados em suas pontas. Quando o ímã passa pelos sensores mencionados, uma corrente resultante da presença eletromagnética exemplificada pela Equação 1 é gerada e passa pelos circuitos do sensor, detectando que o ímã passou por ali naquele instante, indicando que a roda fez um ciclo completo. Com a utilização de mais sensores *hall* pelo limiar da roda, é possível assim detectar a posição que ela está no presente momento, identificando frações de rotação do motor e assim, precisamente, pelo conhecimento da circunferência da roda, ter o cálculo da distância percorrida naquele instante.

Com isso, é possível se saber de forma precisa o quanto a roda andou no momento da detecção e em suas detecções subsequentes, e assim, poder fazer um cálculo cartesiano da posição do robô no espaço, sendo uma das formas mais utilizadas e precisas para detectar a localização do equipamento dentro de uma área controlada.

Após todo o processamento da velocidade pelo ROS, os sistemas da placa e dos *drivers*

de motor ali presentes também fazem a entrega de tensão correspondente aos *hub motors*: isso é, passam a velocidade equivalente para fazer o controle do robô para as rodas, conforme a necessidade do algoritmo de navegação autônoma.

3.2.2 LiDAR

Outro ponto importante visto no robô móvel é a presença de um LiDAR. Sensores LiDAR são ferramentas de escaneamento de ambientes (RAJ *et al.*, 2020), capazes de, em um determinado ângulo, variável de equipamento pra equipamento, determinarem a distância equivalente de cada objeto para o centro do LiDAR, dentro de uma área equivalente na qual o ângulo máximo do aparelho é capaz de detectar. Por exemplo, um LiDAR capaz de detectar uma área de 180 graus vai verificar a distância presente entre si mesmo e o objeto mais próximo para cada parte da angulação correspondente, criando, de fato, um *scanner* do ambiente. Essa descrição do campo de visão pode ser explicada de forma visual pela Figura 6.

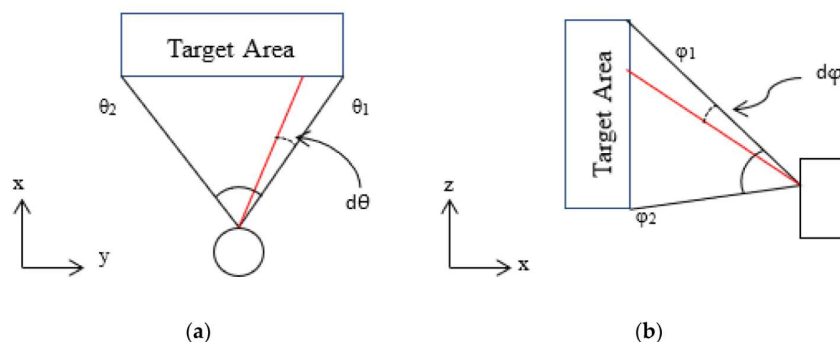


Figura 6 – Campo de visão equivalente de um LiDAR.

Fonte: Thinal Raj (RAJ *et al.*, 2020).

Com isso, pode-se utilizar de diversas aplicações com base nesse funcionamento. Uma aplicação relevante é a detecção de objetos próximos à maquinários pesados, com alta periculosidade (COELHO *et al.*, 2018). Quando um trabalhador tiver distância suficiente de uma máquina em funcionamento, o LiDAR detectará e ativará circuitos internos próprios para efetuar o desligamento da máquina e não por em risco a saúde e segurança do trabalhador em atividade, fazendo aparelhos entrarem adequadamente dentro de regras de segurança NR-12.

O AMPharmy possui sua versão simulada, que foi amplamente utilizada nesse projeto, mas também possui sua versão física, na qual o LiDAR é um SZ-16D, da empresa japonesa Keyence. Seu funcionamento é dado pela detecção da distância via um sensor infravermelho presente dentro de si. A saída da luz infravermelho pela abertura do vidro do LiDAR é dada

pela reflexão de um espelho interno, o qual constantemente gira, permitindo que o laser seja refletido em uma rotação, captando diversos ângulos. As especificações do LiDAR pela Keyence certificam que o equipamento é capaz de pegar um total de 270 graus de leitura, os quais são divididos em 750 unidades de leitura.

Para a captação de dados do SZ-16D, uma conexão RS-422A deve ser lida, a qual parte do equipamento em si. Essa conexão é convertida de acordo com seus valores de dados para uma saída USB regular, a ser decodificada pela Raspberry Pi. Para a leitura de valores, uma entrada hexadecimal deve ser enviada ao LiDAR, que, em retorno, passará a enviar todos os dados de leitura para o sistema. Esses dados são lidos por uma biblioteca Python em conjunto com ROS.

No total, cada leitura corresponde à aproximadamente 1500 dados. Esses dados são enviados em sequências hexadecimais de 16 *bytes*, os quais os 14 primeiros correspondem à uma metade da leitura de distância, que será unificada para ter seu valor total com a leitura do próximo dado, e os últimos dois bytes correspondem aos valores de reflexão do laser e de força do sinal para cada um dos 750 valores.

Se valendo do escaneamento do ambiente pelo LiDAR enquanto se move o robô, é possível fazer o mapeamento de qualquer área no qual ele esteja presente, criando o mapa correspondente do local em questão com as bibliotecas equivalentes no ROS. O mapa completo do ambiente onde o AMPharmy se localiza, feito com o robô móvel juntamente do LiDAR, o qual pode ser visto na Figura 7.

Por meio da triangulação do mapa, fazendo a correspondência da leitura vista pelo LiDAR naquele instante e do que preenche e corresponde à imagem, o robô móvel é capaz de se localizar em qualquer ambiente, visto que possua uma equivalência entre a leitura do *scanner* e do mapa.

Também é importante se mencionar que, na manipulação do robô criado no ambiente simulado, não apenas um LiDAR foi utilizado, mas uma construção envolvendo cinco sensores LiDAR, colocados de forma paralela no corpo do equipamento, verticalmente, buscando conseguir uma leitura não apenas restrita à um plano bi-dimensional, mas múltiplos, com alturas diferentes. Com a junção da leitura de cinco *lasers*, é possível se criar uma leitura similar à de um ambiente tri-dimensional, onde a coleta de informações de cada um dos sensores é sobreposta à outra, criando uma dimensão topográfica com suas detecções, aumentando a precisão da leitura com mais dados.

A utilização de cinco sensores para a leitura de dados deve-se não apenas à tentativa



Figura 7 – Mapa completo do Habitat Senai.

Fonte: O Autor.

de simular uma ambientação tri-dimensional, mas principalmente à detecção de objetos independente de sua altura dentro do universo simulado. Uma placa, por exemplo, possui uma base fina e um encorpamento mais grosso: algo assim acaba por confundir um robô de transporte automaticamente navegável, especialmente quando se considera que o AMPharmy possui uma base relativamente alta. A Figura 8 mostra os cinco LiDARs no robô simulado.

3.2.3 Câmera

Para conseguir a leitura de imagens ao redor do robô, se é usada uma câmera padrão genérica em uma porta USB do sistema. O sistema operacional possui integração nativa com a câmera, o que a torna fácil e simples de usar.

No ambiente simulado, a utilização da câmera também é possível, e de fácil execução. Uma câmera foi adicionada ao corpo do robô virtual dentro do *software* CoppeliaSim, e, por meio da execução do código Lua, publicada dentro do tópico `/usb_cam/camera_info`, para os metadados da imagem, como altura e largura do *frame*, rotação, e tudo o que pode ser incluso dentro de uma mensagem do tipo `sensor_msgs/CameraInfo`.

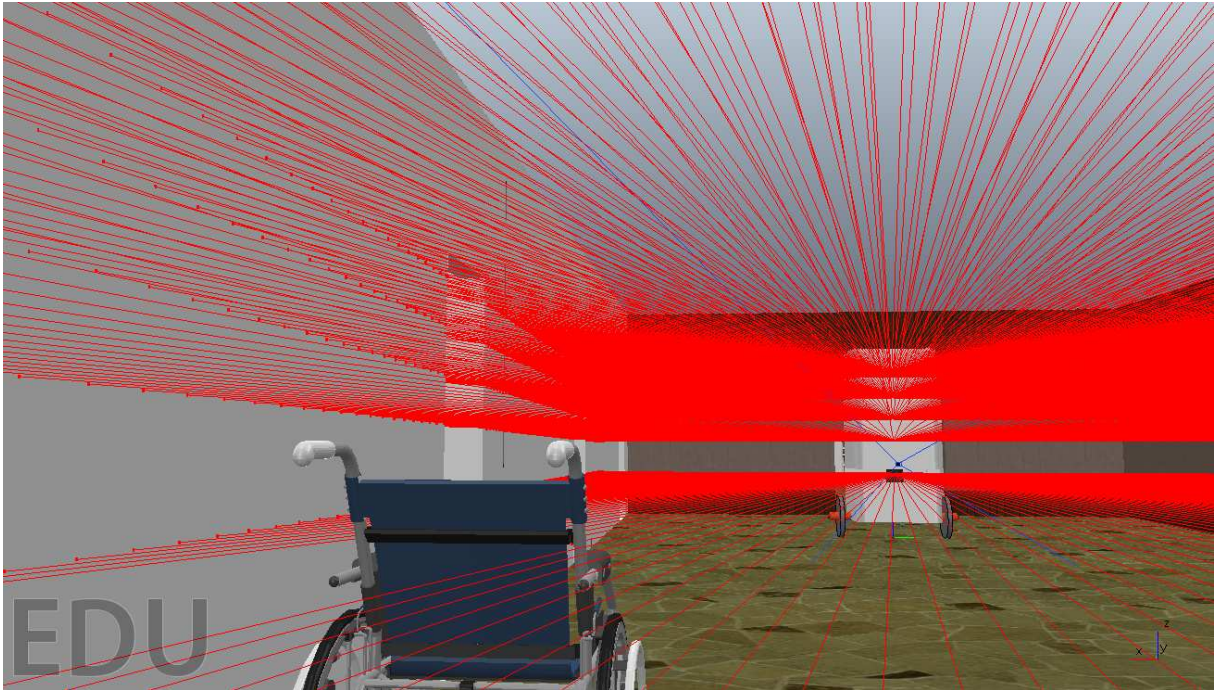


Figura 8 – Robô simulado com cinco lasers.

Fonte: O Autor.

O código Lua também publica em um segundo tópico, `/usb_cam/image_raw`, que é capaz de publicar a própria imagem da câmera, para fins de auxílio na visualização do que o robô está vendo com a câmera nesse momento. Um exemplo da saída desse tópico pode ser visto na Figura 9, que mostra a visualização da imagem pura após ser processada pelo OpenCV.

Com a utilização da câmera, se torna possível utilizar todas as funcionalidades do OpenCV, podendo habilitar uma navegação autônoma especial, onde, por meio de processamento de imagens, é possível detectar *labels* correspondentes aos objetos ao redor do ambiente, real ou simulado, e colocar marcadores no mapa com suas posições.

3.3 O ROBÔ REAL

Para a execução do projeto, o robô móvel AMPharmy foi usado, construído pela *startup* 33Robotics, localizada em Curitiba, no Paraná. O robô original foi criado para ser um robô autônomo, capaz de entregar medicamentos dentro de ambientes hospitalares e leitos clínicos. O robô pode ser visto mais claramente na Figura 10.

O robô móvel AMPharmy real foi desenvolvido, testado e validado nos corredores do Habitat Senai, em Curitiba, onde a startup 33Robotics está localizada. Não obstante, o robô também foi testado e validado dentro de hospitais na cidade. Ele é capaz de rodar com um sensor LiDAR SZ-16D, desenvolvido pela empresa japonesa Keyence, fazendo o sensoramento à *laser*

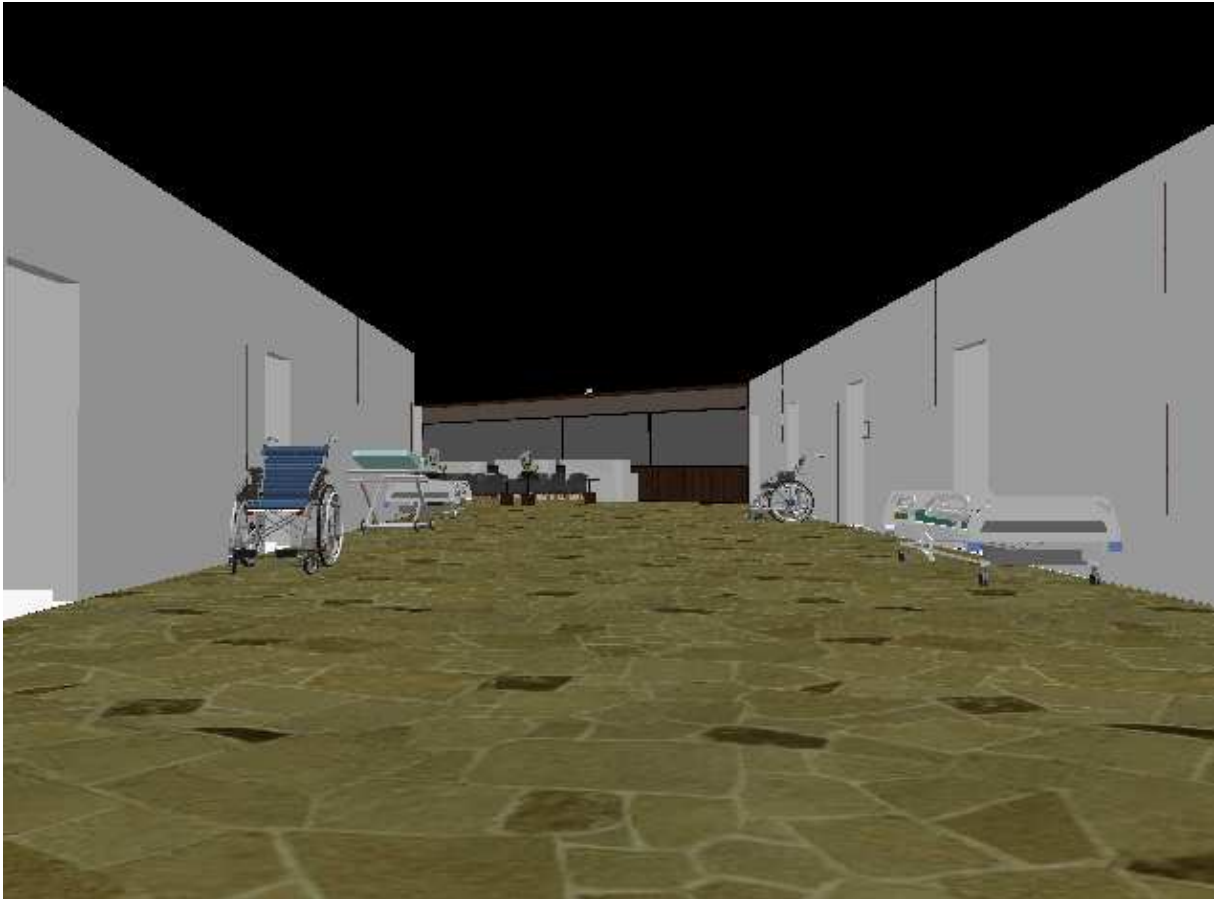


Figura 9 – Frame da câmera vinda do tópicos correspondente no ROS, pós processamento no OpenCV.

Fonte: O Autor.

de todo o ambiente ao seu redor. O AMPharmy físico também possui rodas com *hub motors*, ajudando na sua odometria, e todo seu processamento com o ROS é executado dentro de uma Raspberry Pi 4, rodando um sistema operacional Ubuntu 16.04.

O robô Pharmy real também opera em uma velocidade aproximada de 0.4 metros por segundo. A legislação brasileira permite uma velocidade de até 1.5 metros por segundo em operação, porém, foi escolhida uma velocidade melhor tanto buscando enfatizar a segurança do projeto, para evitar riscos e acidentes, mas também buscando melhorar a qualidade da navegação.

Porém, buscando um projeto mais eficiente que pode ser testado em várias ocasiões sem a necessidade atual do robô, uma versão digital do projeto foi modelada, projetada e criada dentro de um software de simulação.

3.4 O ROBÔ SIMULADO

A versão simulada do AMPharmy foi construída dentro do software de simulação CoppeliaSim. Com o CoppeliaSim, é possível criar no projeto um robô capaz de imitar um



Figura 10 – AMPharmy, em navegação pelos corredores do Habitat Senai.

Fonte: Autor.

equipamento real, considerando seu tamanho, peso, formato, moção das rodas, sensores, câmeras e mais. Essa versão simulada da máquina é equipada com um sensor de odometria inteiro, capaz de detectar o quanto o robô se moveu em relação a seu espaço base. Também tem em si equipado cinco sensores LiDAR, capazes de escanear o ambiente e construir mapas dele ao redor de si próprio, e uma câmera, que é capaz de detectar e ampliar o conhecimento do universo ao seu redor por visão computacional.

Para esse projeto, um mapa 3D do hospital foi construído, possuindo objetos e salas diferentes para o robô identificar por meio da visão computacional, além de um *design* simplificado do AMPharmy para representar o equipamento. Os principais pontos das cenas do espaço simulado são uma sala de recepção, uma sala para a farmácia, uma sala para o paciente e um corredor conectando todos esses ambientes, no qual pode-se localizar cadeiras de roda e macas. Todas essas cenas podem ser vistas na Figura 11.

A versão simulada do AMPharmy pode ser vista na Figura 12. Ela mantém o *design* original do robô, visto nas Figuras 3 e 10, adicionando seus principais sensores fixados no corpo do robô no software do Coppeliasim. Para fazer com que o robô seja capaz de navegar

automaticamente e manualmente, conectando as mensagens correspondendo com o que vem dos sensores, o *framework* do ROS foi instalado e configurado. Como o sistema operacional em uso no projeto é o Ubuntu 20.04, a versão do ROS em uso no projeto é a equivalente do sistema, isso é, o ROS Noetic Ninjemys, que também é a última versão lançada do ROS antes da migração completa dos desenvolvedores para o ROS2.

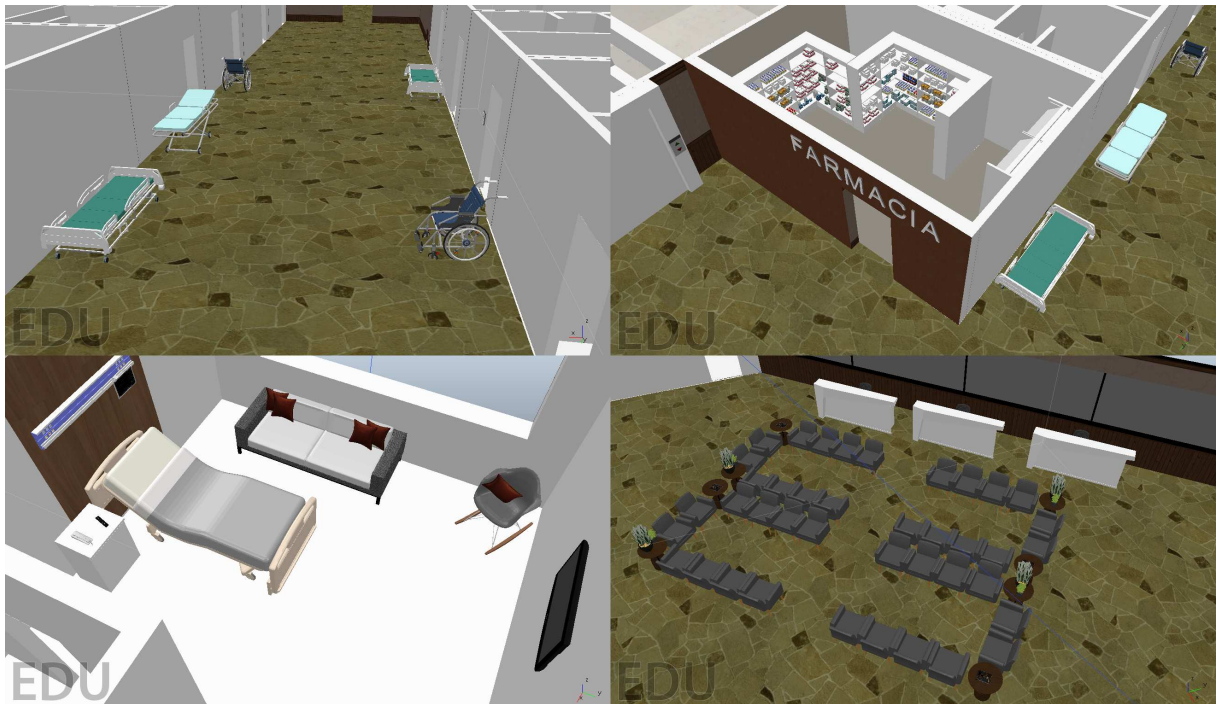


Figura 11 – As principais salas do mapa, criadas dentro do software de simulação CoppeliaSim.

Fonte: Autor.

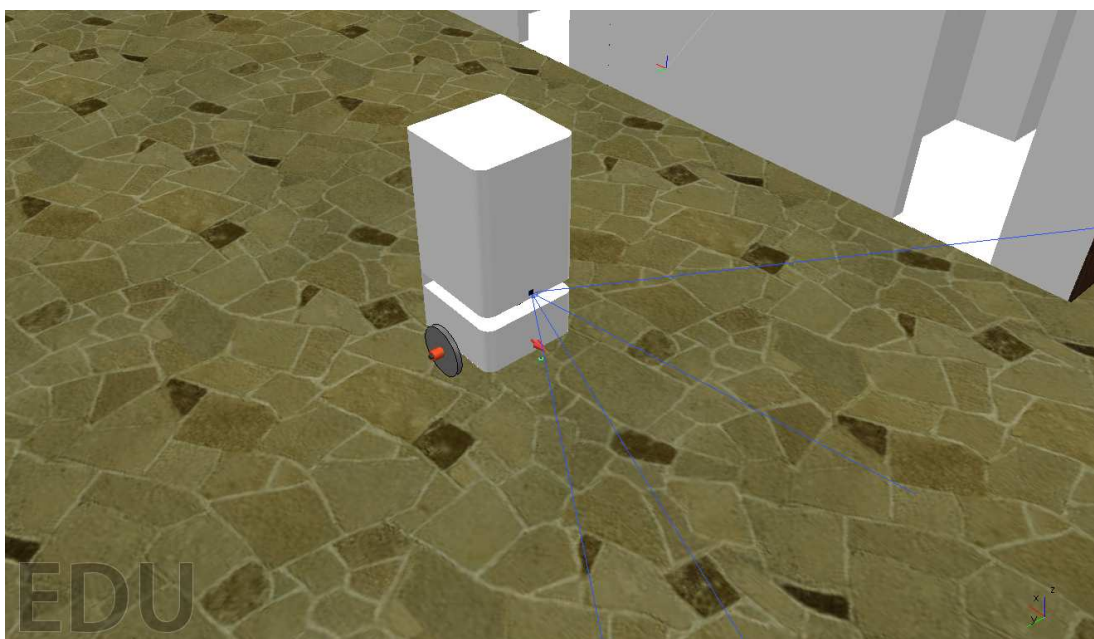


Figura 12 – A versão simulada do AMPharmy.

Fonte: Autor.

Para isso, as partes principais do corpo do robô são configuradas em um código Lua, que pode mandar e interpretar mensagens pelo ROS. A velocidade do robô, que é dada pelo tópico *cmd_vel*, pode ser lida pelo código Lua, sendo interpretada pelos seus valores lineares e angulares, e providenciando, a partir desses valores, a velocidade correspondente correta para o robô virtual, permitindo-o navegar e mover-se pelo ambiente simulado.

O robô recriado também pode mandar mensagens de informações lidas pelos seus sensores. Enquanto o comando *roscore* estiver ativo, o simulador CoppeliaSim vai publicar automaticamente todos os tópicos que um robô real publicaria, como sua odometria, leituras do laser, transformadas, leituras da câmera, entre outros.

4 PERCEPÇÃO VISUAL

Buscando criar um modelo que seja capaz de reconhecer imagens de maneira satisfatória dentro de um ambiente, se valendo de visão computacional, deve-se construir primeiramente um projeto capaz de processamento de imagens, lendo seus dados por uma câmera. A construção desse modelo e do *dataset* equivalente, então, foi a primeira parte de desenvolvimento desse projeto, se valendo de um robô real, o AMPharmy, para a construção dessa etapa.

Para o modelo de processamento de imagem e leitura de dados para a localização por câmera, foi utilizada uma câmera USB padrão, capaz de detectar os frames necessários e fazê-los passar pelo processamento da YOLOv8.

Para a utilização da YOLOv8, porém, o autor da dissertação se valeu da criação de um novo *dataset*, customizado com as particularidades encontradas no ambiente de *co-working* do Habitat Senai. Com isso, o *dataset* customizado é capaz de pegar informações mais relevantes para a área em questão, onde suas *labels* particulares são detectadas pelo sistema.

Por esse passo, 995 fotos de vários pontos do ambiente foram tiradas por todo o lugar. Essas fotos foram separadas em vinte e duas *labels* diferentes para serem treinadas de forma correta. Isso foi possível com a aplicação do *website* Roboflow, que pode ser visto em mais detalhes no trabalho de (DWYER; NELSON, 2022). A página da internet deixa possível que, com a utilização de uma conta, qualquer usuário seja capaz de criar *labels* em fotos de forma fácil e simples, mesmo com uma gama vasta e variada de imagens.

Para isso funcionar, o processo precisa começar de algum ponto. Pelos passos do Roboflow, todas as 995 imagens precisaram ser categorizadas, uma a uma, em todas as *labels* as quais elas foram correspondentes, criando *bounding boxes* para cada uma delas. *Bounding boxes* são artifícios comumente utilizados quando se trata de detecção de objetos, especialmente para representar de forma gráfica os objetos classificados em determinadas imagens após seu processamento. Elas são representadas, literalmente, pela construção de um quadrado ao entorno de uma figura que foi reconhecida pelo processamento da imagem, e após envelopar esse quadrado, tradicionalmente a *label* identificada e sua taxa de confiabilidade na predição são colocadas acima do objeto.

Para o trabalho atual, as 995 imagens foram todas as imagens que continham um ou mais *labels*. A quantidade total, contando também as que não foram categorizadas em *label* alguma, chega em um valor de 1169 fotografias. Essas fotos foram deixadas sem categoria por

diversos motivos, seja ele por baixa qualidade da imagem, seja porque a imagem não obtinha informação alguma, como pode ser visto na Figura 13, seja por obstrução do objeto que deveria estar em evidência, entre outros fatores.



Figura 13 – Uma foto sem informações relevantes.

Fonte: O Autor.

O *website* do Roboflow também tem suporte para *data augmentation*. Esse é um processo que implica no aumento artificial de imagens do *dataset* pela sua modificação inerente de seus atributos, com base nas imagens já existentes previamente. A sua eficiência pode ser vista em mais detalhe no trabalho de (PEREZ; WANG, 2017). Essas transformações podem ser geométricas, por mudança de cores, deleção de partes aleatórias da imagem, entre outros aspectos. Alguns exemplos de imagens que passaram artificialmente por *data augmentation* podem ser vistos na Figura 14.

Nesse conjunto de imagens, é possível verificar e analisar vários métodos de *data augmentation*. A primeira imagem da Figura 14 já é um exemplo claro, onde a fotografia recebeu um pequeno zoom e foi modificada para se inclinar para a esquerda. A segunda e a terceira imagem possuem recortes de outras fotos no meio delas. Há exemplos como a décima-segunda imagem, que é quase uma montagem de várias outras em uma só, com tamanho reduzido. Todas

essas modificações são incluídas no *dataset* final, artificialmente aumentando seus números e gerando mais dados para a construção do modelo.

A versão gratuita do Roboflow, por mais que permita a utilização de *data augmentation*, a permite de uma forma consideravelmente restrita. Porém, mesmo com a versão gratuita, vários tipos de modificações podem ser feitas. Mudança em seu tamanho, espelhamento, rotação, e outras várias como as vistas na Figura 14.

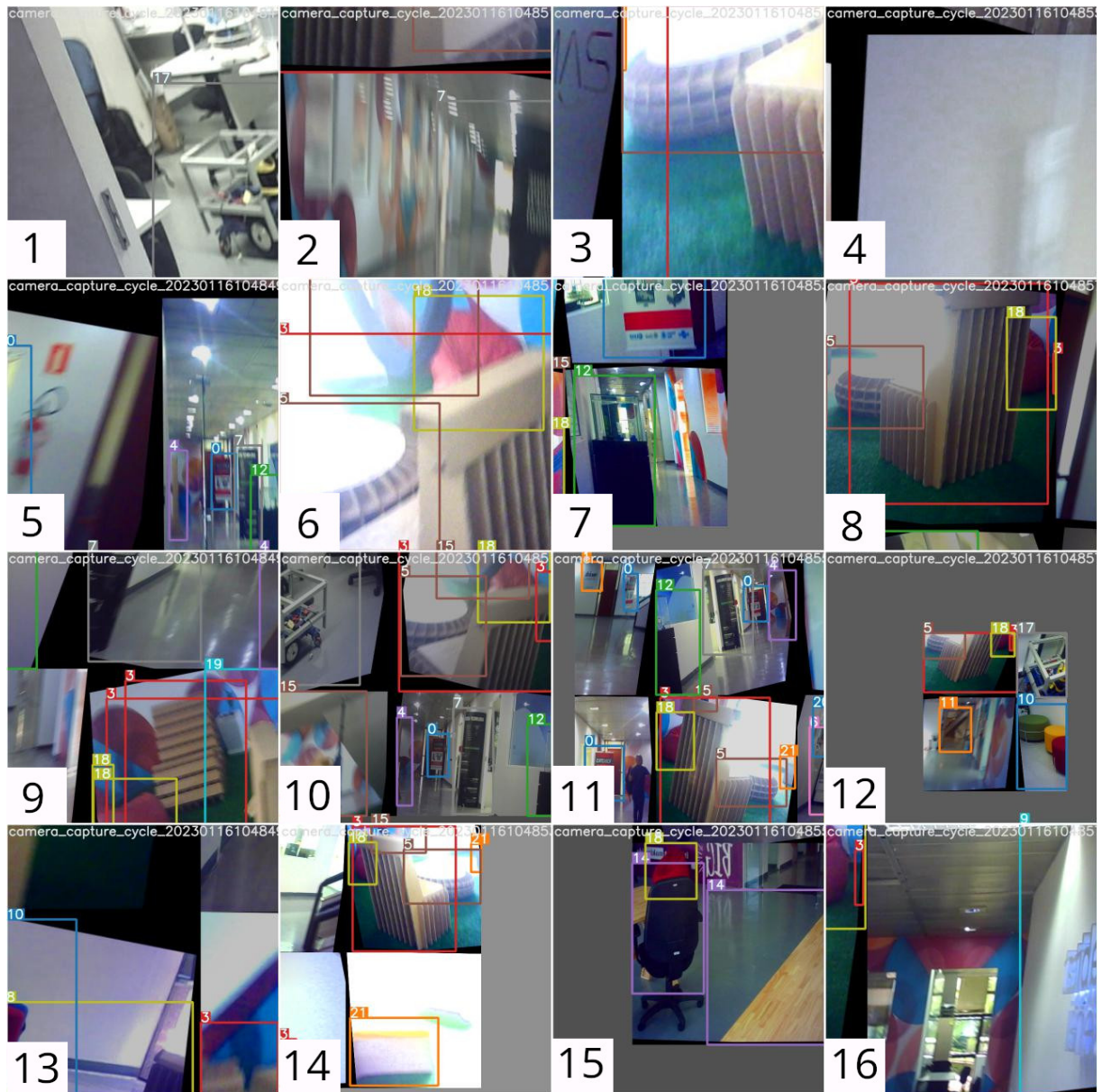


Figura 14 – Imagem com vários exemplos de *data augmentation* com o *dataset* utilizado.

Fonte: O Autor.

Uma vez que todas as imagens já estão classificadas e divididas em seus respectivos *labels*, elas precisam ser processadas de uma forma correta. O primeiro passo para isso é a exportação correta do *dataset* customizado que se foi gerado. Pela utilização de uma *API Key* e

com os detalhes do *workspace* do Roboflow, o trabalho é capaz de começar. Nesse cenário, a próxima etapa é o treinamento do modelo com uma biblioteca num *notebook* Jupyter, antes de ser implantado na API que faz um escalamento automático e que é usada para a exportação de todo o sistema.

Com isso feito, um modelo precisa ser escolhido para ser usado como base. No cenário desse projeto, o modelo que foi selecionado foi o da YOLOv8: mais especificamente, o modelo de exportação da YOLOv8 com PyTorch para um treinamento customizado. A forma que essa utilização simplifica todo o treinamento de forma drástica é pelo uso da estratégia de *Transfer Learning*.

Esse plano de ação, em resumo, coleta todo um grande *dataset* que já existe como base e transfere o que já se foi aprendido com ele para treinar de forma simplificada um novo *dataset*, de forma rápida e precisa. No caso, o *dataset* que foi usado como base é o *dataset* da YOLOv8. O treinamento total de toda a criação de um novo modelo por *Transfer Learning*, utilizando as 995 imagens, levou menos de duas horas, com resultados precisos. Também acaba por ser menos custoso computacionalmente, fazendo desse método extremamente útil quando necessário para treinar um *dataset* com uma quantidade pequena de imagens: o que é exatamente o caso desse projeto.

A execução correta do modelo de *Transfer Learning* depende de uma série de etapas baseadas em um modelo de Redes Neurais. Após escolher o modelo base, no caso, a YOLOv8, as suas *layers* iniciais de treinamento serão todas congeladas. A ideia do "congelamento" de *layers* é que elas não serão modificadas com o andamento do projeto. Elas sempre vão reter os seus parâmetros iniciais, os seus pesos. Com isso em mãos e executando, o modelo evitará de acabar sendo re-treinado por inteiro de novo, sem necessidade, e assim mantendo suas características essenciais que serão utilizadas pelo modelo novo.

Após isso, a informação nova precisa ser adicionada, após os *layers* que foram congelados, entrando no topo das *layers* remanescentes da *network* base de referência. Como isso gera um resultado novo, esse modelo precisa ser treinado novamente, mas agora retraindo a informação inicial que o modelo base possuía de forma praticamente perfeita, como se realmente estivesse congelada num sentido literal, sendo usada como parâmetro para re-treinar as informações novas de maneira rápida, precisa e eficiente.

No final do processo, uma certa afinação para maior precisão de dados é necessária, fazendo com que o treinamento seja repetido algumas vezes, descongelando partes diferentes dos

layers iniciais, usando taxas de treinamento mais lentas e fazendo algumas alterações mínimas. Tudo isso é feito com o intuito de gerar modelos diferentes, que serão comparados entre si e analisados para ver qual deles gera um resultado melhor e com maior acurácia. Essa afinação pode ser vista com uma clareza mais visual na Figura 15.

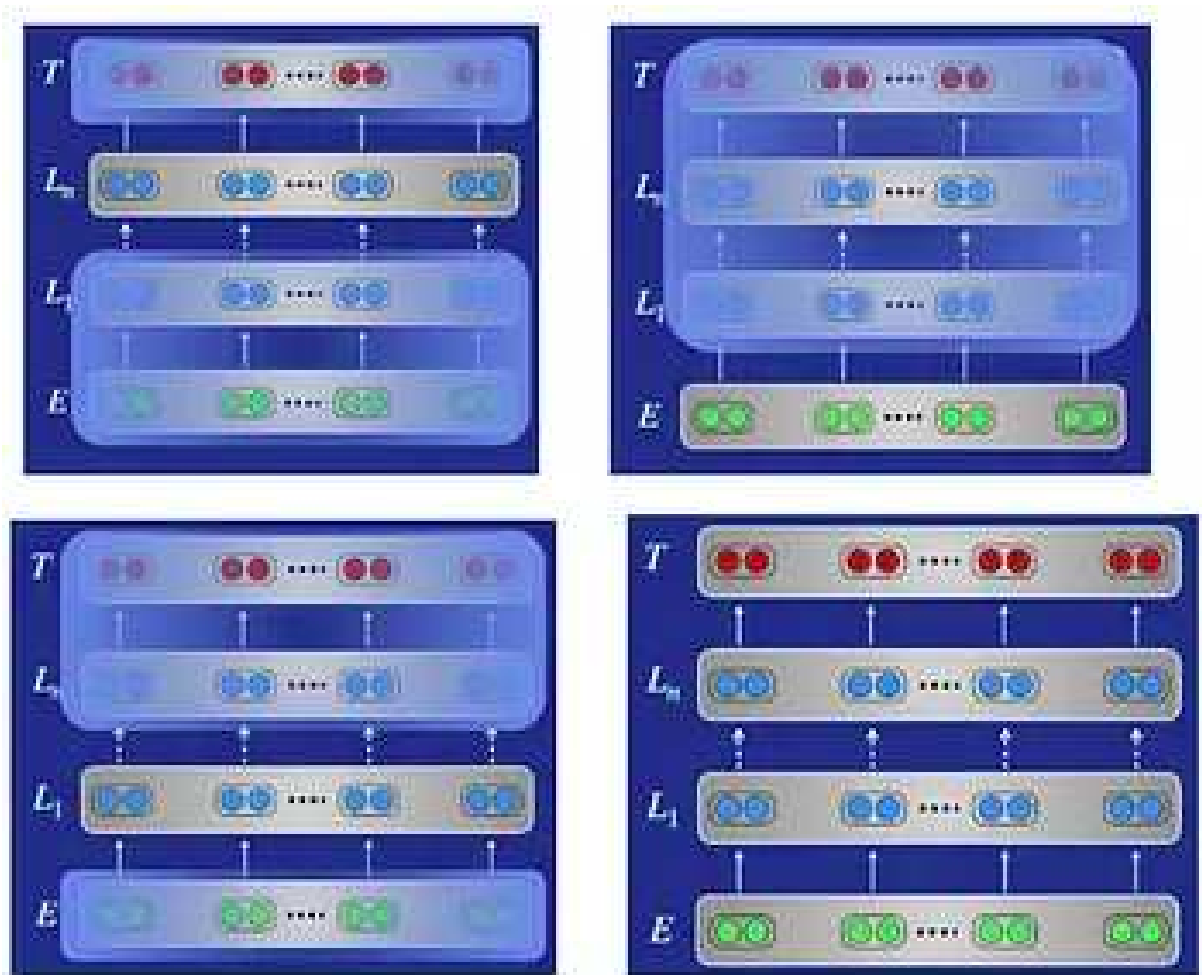


Figura 15 – Exemplo de descongelamento de *layers* diferentes, com taxas de aprendizado diferentes, para geração de resultados diferentes.

Fonte: Pratik Bhavsar (BHAVSAR, 2019).

Isso se dá bastante ao trabalho feito por (FELBO *et al.*, 2017), onde se é dito que ao invés de simplesmente congelar todas as *layers* exceto a última, o mais eficiente é ir descongelando elas gradativamente para manter elas todas capazes de serem treinadas, mas ainda retendo a informação inicial.

O modelo foi treinado com 55 épocas diferentes. Cada nova época gera um modelo teoricamente mais avançado que o anterior, pegando das falhas de seus antecessores e acrescentando ou retirando o que for necessário para, no final, se ter um aprendizado mais efetivo e resultados mais precisos. Cada época diferente também traz as alterações diferentes no congelamento dos

layers, onde o modelo aprende o que acaba por ser mais efetivo e o que é contra-intuitivo na criação do projeto final.

Para construir o modelo de visão para a detecção de objetos, primeiramente, é necessário coletar as imagens do ambiente desejado, e apenas então adicionar as *labels*, imagem por imagem. A *label* em uma detecção de objetos consiste de um quadrado desenhado ao redor do objeto detectado, a chamada *bounding box*. Uma porcentagem da acurácia da predição também pode ser vista acompanhada do *label*. Um exemplo de uma *bounding box* pode ser visto na Figura 16.

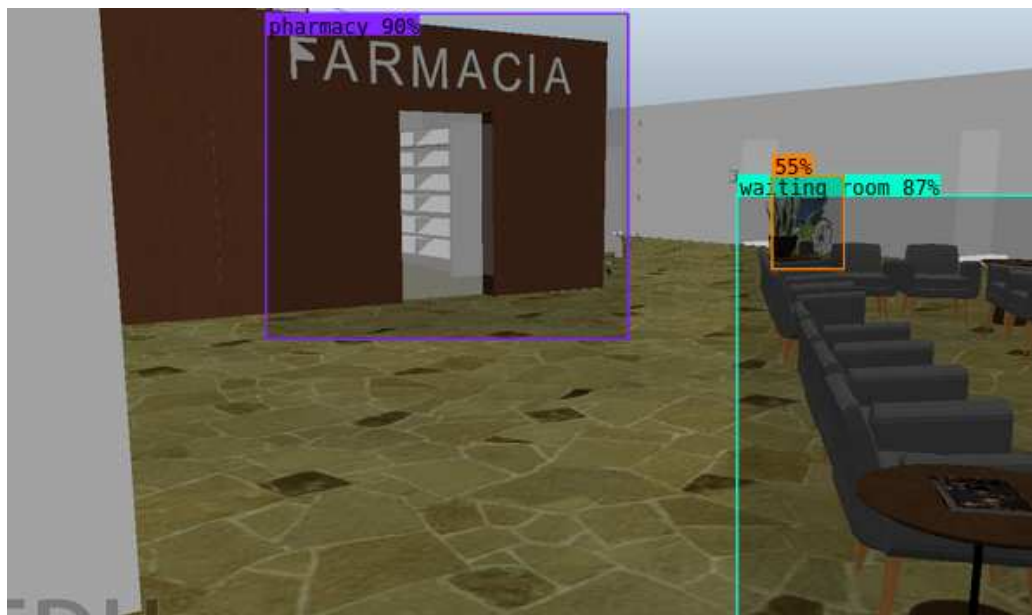


Figura 16 – Objetos reconhecidos, com suas *bounding boxes* correspondentes.

Fonte: O Autor.

Usando os modelos de *Transfer Learning*, o objetivo final é fazer um modelo novo para a detecção de objetos, utilizando o modelo antigo como base. Dessa forma, a maior parte do conhecimento do método base será mantida. Porém, o conhecimento adquirido pelo método novo será transferido para os *layers* do método base, fazendo um modelo novo com resultados consistentes, tomando um tempo consideravelmente menor para ser construído, gerando um *dataset* completo de forma rápida e eficiente.

Para que o *Transfer Learning* funcione, um modelo de referência deve ser utilizado com a nova base para adquirir conhecimento. Esse modelo pode ser retrabalhado com uma ampla variedade de possibilidades. Contudo, para o trabalho neste projeto, o escolhido foi o modelo YOLOv8 da Ultralytics, uma tecnologia de ponta para detecção de imagem. Utilizando um *notebook* do Google Colaboratory como referência, fornecido pelo modelo Roboflow em parceria com a Ultralytics.

Para uma versão inicial do conceito, foi-se preparado um *dataset* com a versão física do robô AMPharmy, nos corredores do Sistema FIEP, sendo coletadas 996 fotos. No final, essas fotos foram divididas em setenta, dez e vinte por cento para os grupos de treinamento, teste e validação, respectivamente. No entanto, com todo o trabalho de aumento de dados feito no modelo, o tamanho de seu grupo de treinamento foi aumentado de 696 para 2100 fotos, resultando em uma nova divisão entre os três grupos de oitenta e sete por cento para teste, quatro por cento para treinamento e oito por cento para validação. Essas imagens foram aumentadas por meio do uso de viradas horizontais, rotação entre trinta graus e distorções, tanto horizontais quanto verticais.

Isso pode ser visto, por exemplo, na Figura 17, onde uma dessas fotos do grupo de teste pode ser observada. Essa foto foi tirada em um corredor dos escritórios do Habitat Senai, mostrando vários *banners* representando todas as empresas localizadas naquele corredor. No caso dessa imagem, todas as empresas ali posicionadas, seus logotipos e *banners* foram completamente reconhecidos pelo algoritmo do modelo, ajustando corretamente cada um às etiquetas disponíveis. Além disso, a imagem também mostra que ele reconhece placas de papelão posicionadas nas bordas laterais do canto da figura.

Após a verificação do modelo no mundo real, se passou para o ambiente simulado, onde o AMPharmy virtual faria os testes para gerar os resultados buscados por essa dissertação de mestrado. Para isso, se utilizou a câmera virtual instalada dentro do ambiente do CoppeliaSim.

Foram criados nove rótulos *labels* no projeto, cada um com um nome indicativo do que representam. "Bed" foi criado devido à necessidade de entregar medicamentos aos pacientes conforme necessário, e, por razões semelhantes, "Large Stretcher" e "Small Stretcher" foram criados, com a diferença de serem objetos móveis, mas servindo ao mesmo propósito. Seu reconhecimento na câmera serve como um auxílio visual para o robô evitá-los durante o deslocamento. O mesmo pode ser dito para "Wheelchair", que tem um propósito semelhante, mas para cadeiras de rodas em vez de macas. "Pharmacy" foi criado para que os medicamentos adequados pudessem ser adquiridos para as entregas, enquanto "Medicine" refere-se às prateleiras onde estão localizados. "Elevator" se deve à necessidade de o robô entregar os medicamentos em andares superiores ou inferiores, e "Waiting Room" foi destinado à entrega de documentos para as pessoas que trabalham com pacientes na recepção. O último rótulo, "Sofa", é usado como um meio de referência para o rótulo "Bed", dado que o universo simulado construído sempre tem um sofá ao lado da cama. Portanto, o reconhecimento de um pode também auxiliar na entrega de



Figura 17 – Exemplo de uma figura no grupo de testes.

Fonte: Autor.

medicamentos ao paciente dentro de um quarto.

Um agrupamento de imagens representando a detecção de cada uma dessas camadas, enquanto navega pelo universo simulado, é algo que pode ser observado na Figura 18.

4.1 RESULTADOS

Nessa seção, serão discutidos os resultados obtidos com a parte referente à percepção visual, sobre o que se foi obtido quanto ao algoritmo desenvolvido para a detecção de objetos.

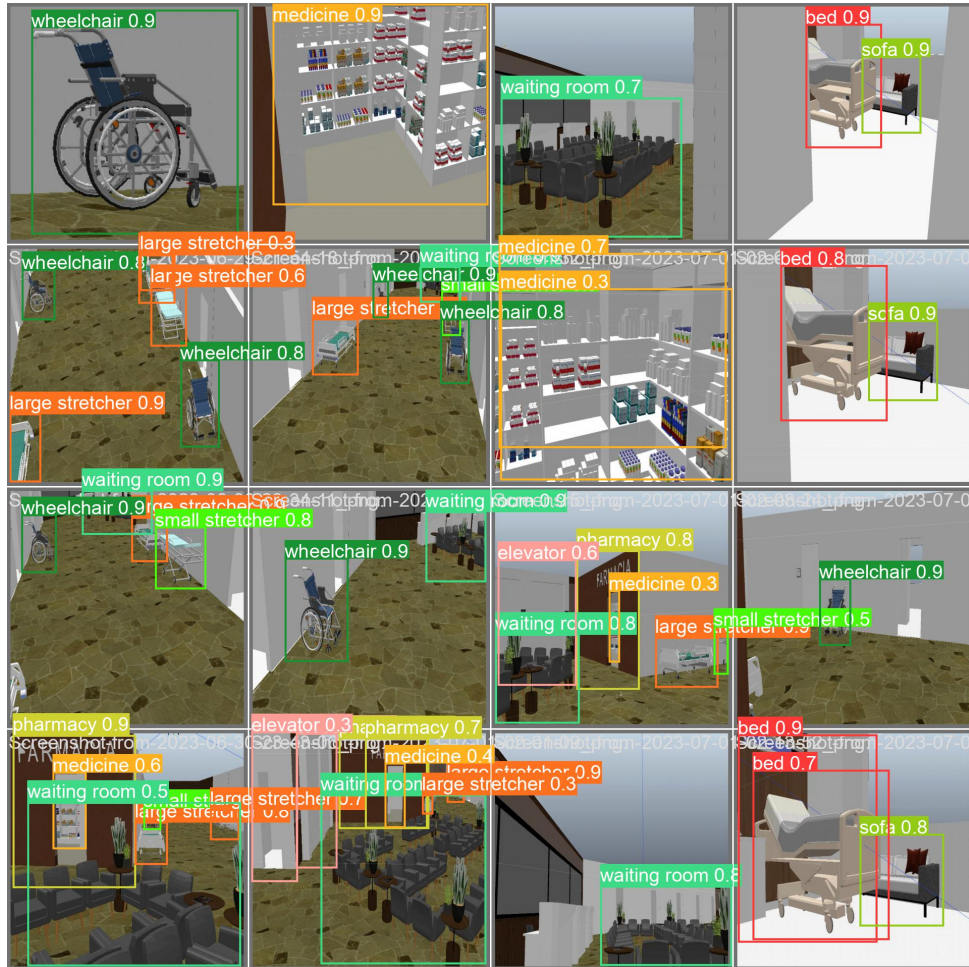


Figura 18 – Várias imagens mostrando os *labels* reconhecidos no universo simulado.

Fonte: Autor.

4.1.1 Da Detecção de Objetos

Os resultados podem ser mensurados de diferentes maneiras. O conjunto de dados foi treinado com 30 épocas, utilizando um total de 199 imagens após a realização de *data augmentation* para os nove *labels*. O resultado do modelo treinado na *Transfer Learning* pode ser visualizado na Figura 19, que representa sua Matriz de Confusão.

A interpretação da Matriz de Confusão é bastante simples (VISA *et al.*, 2011). Os valores na diagonal principal representam um valor numérico para quão preciso é o modelo: quanto mais próximo o valor estiver de 1 e, na representação da Figura 19, quanto mais escuro for o azul, mais preciso é aquele rótulo específico.

Neste caso, todos os rótulos têm um nível de precisão quase perfeito, exceto para ambos os rótulos de macas. Isso significa que sempre que os outros rótulos estão na imagem, eles são reconhecidos corretamente; sempre que não estão, o modelo não os indica. Portanto, a quantidade de Falsos Positivos e Falsos Negativos é consideravelmente baixa, indicando que o

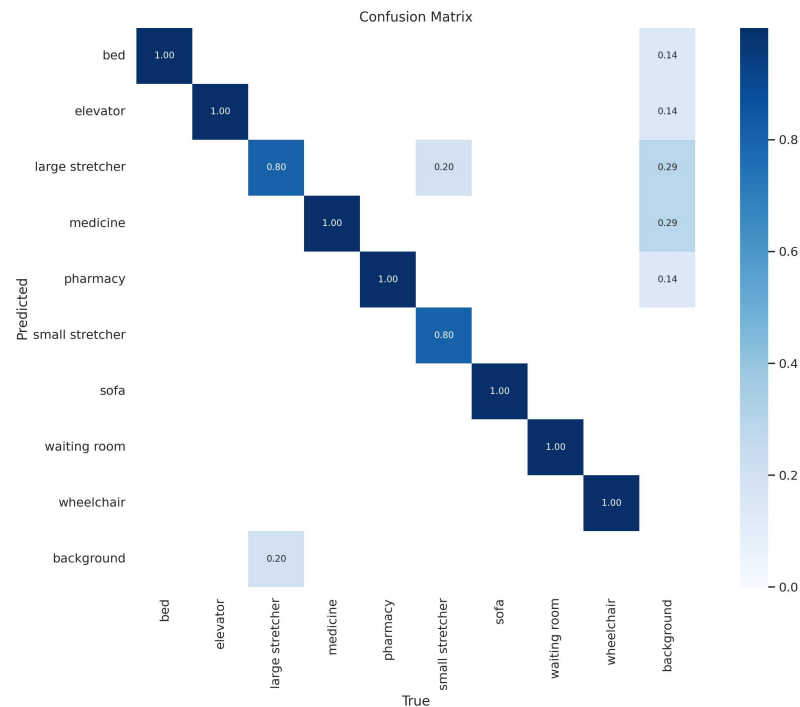


Figura 19 – A Matriz de Confusão do modelo novo.

Fonte: Autor.

modelo fornece um bom resultado.

Esses resultados também podem ser medidos por outros meios, como o *Recall*, *Precision* ou as curvas mAP (BUCKLAND; GEY, 1994), como visto na representação da Figura 20.

Para os valores observados nestes gráficos, os valores das métricas de *Precision* e *Recall* podem ser facilmente identificados. Cada passo do eixo x corresponde a uma época em que o modelo foi treinado, portanto, dado que foram utilizadas 30 épocas no treinamento do *Transfer Learning*, o eixo x é composto por 30 etapas.

Quanto maior o valor de *Precision*, maior o valor lido de positivos em comparação com o valor real de positivos. Isso significa que o sistema pode prever melhor os positivos que são realmente positivos, que representa aproximadamente a porcentagem de vezes que o modelo está correto. Quanto ao *Recall*, esse é o valor de verdadeiros positivos sobre o número total de vezes que o rótulo existe nas imagens. Isso significa que o sistema pode prever melhor a presença de um *label* na imagem. Ambos os valores podem ser uma indicação de quão bom é o modelo, pois se aproximam de 1 (GOUTTE; GAUSSIER, 2005).

O mAP funciona de maneira semelhante. O valor do mAP é dado pela mediana da Precisão Média, encontrada pelo inteiro do gráfico de *Precision*, ou seja, a área sob ele. O valor do mAP também é uma referência para um sistema mais preciso quanto mais alto ele for, se aproximando de um valor de 1. Para isso, muitos mAPs diferentes podem ser calculados com

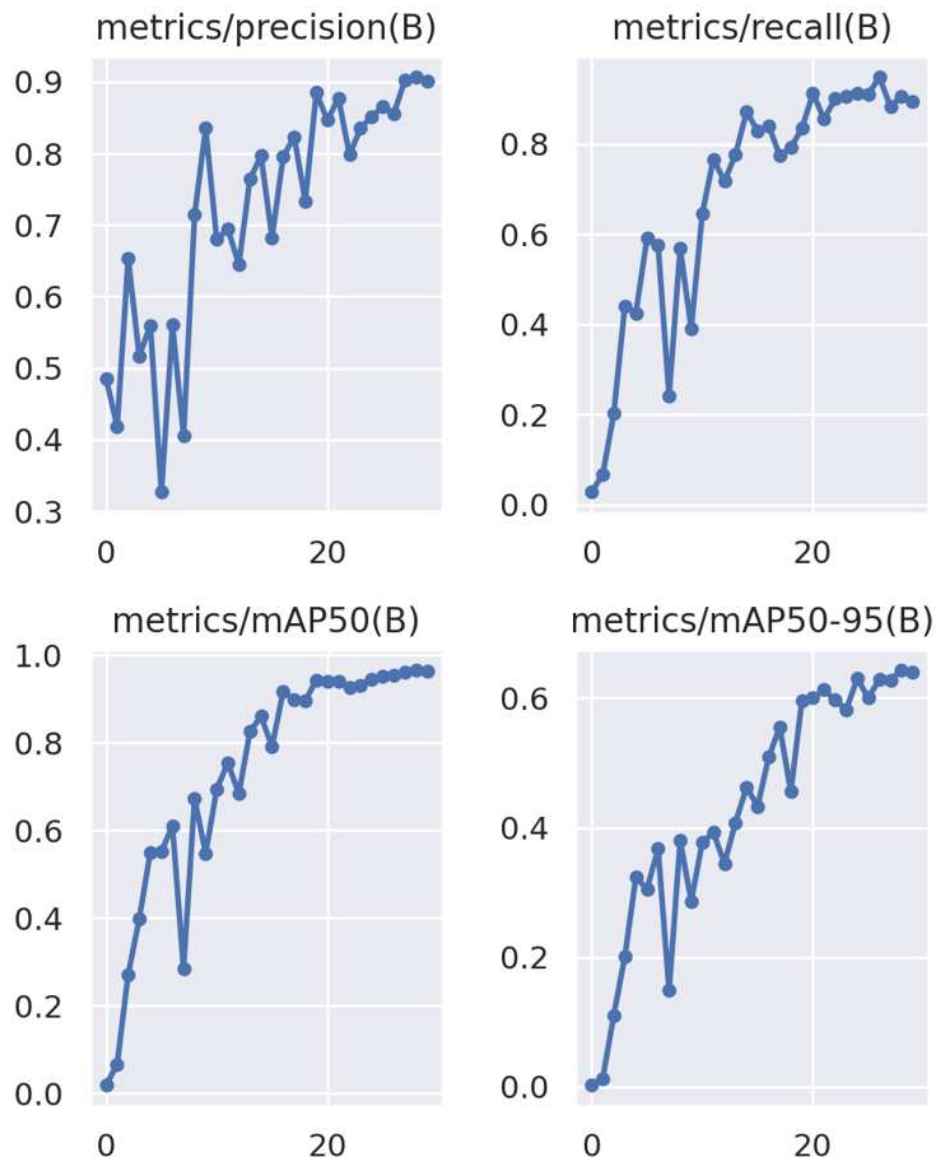


Figura 20 – Curvas representando os resultados do modelo.

Fonte: Autor.

diferentes *Intersections over Union* (IoU), o nível de um determinado limiar para uma *bounding box* considerar uma imagem reconhecida como o rótulo pretendido.

Para o exemplo na Figura 20, o IoU foi considerado como 0,5, o que é equivalente a uma certeza mínima de 50% em cada rótulo quando reconhecido. Os valores de mAP são consideravelmente altos, ultrapassando 0,9 mesmo antes de atingir 20 épocas. Os níveis de *Precision* e *Recall* também são igualmente bons, chegando em valores próximos de 1. A *Precision* atinge 0,9 nas últimas três épocas, e o *Recall* mantém um valor estável acima de 0,8 para todas as medições além da vigésima época.

Se valendo dos resultados obtidos pelos gráficos gerais de mAP50, além das curvas de

Precision e Recall do modelo, pode-se afirmar que ele possui uma precisão consideravelmente alta, podendo garantir sua confiabilidade na execução do projeto referente.

Essa análise e estudo sobre percepção visual também foi publicada e apresentada no Simpósio Latino-Americano de Robótica (LARS) de 2023 pelo autor, cujo artigo pode ser visto em (FABRICIO *et al.*, 2023a).

5 IDENTIFICAÇÃO DE DISTÂNCIA VIA CÂMERA

O objetivo do projeto AMPharmy é criar a base para um sistema que é capaz de navegar por um hospital, entregando objetos diversos por uma área geral, buscando reduzir o tempo no qual médicos e enfermeiros precisam gastar indo de um local ao outro e permitindo-os ficar mais tempo com seus pacientes.

O maior objetivo do robô, nesse caso, é a entrega de medicamentos dentro do hospital, principalmente da farmácia para o leito, onde o paciente está. Porém, ele também pode ser usado para a entrega de outras coisas, caso necessário. Como parte do projeto AMPharmy, essa dissertação apresenta o algoritmo de visão computacional utilizado para identificar objetos e estimar sua posição.

5.1 ALGORITMO PROPOSTO PARA O CÁLCULO DE DISTÂNCIA

É necessário identificar a distância do objeto em relação ao robô. A maioria dos trabalhos relacionados utiliza sensores 3D para essa função, porém, o uso desses sensores gera um custo computacional mais elevado. Este trabalho concentra-se em identificar a distância do objeto do AMPharmy por meio da câmera.

A abordagem consiste em inferir informações prévias, como o tamanho de objetos específicos no ambiente hospitalar. Tendo o tamanho real desses objetos, é possível calcular a distância do objeto em relação à câmera, o que é suficiente para identificá-lo. Dessa forma, a abordagem consiste em identificar o objeto, procurar seu tamanho em um banco de dados e, em seguida, calcular sua posição (distância entre o objeto e o robô) no ambiente.

Todos os objetos têm um valor fixo (tamanho), onde uma medição foi realizada, descobrindo seus comprimentos a uma distância de 2,5m da câmera do robô. Com esses valores sendo estabelecidos como referência, onde todas as *labels* possuem um valor, salvos em um arquivo .json de fácil acesso, pode-se utilizar uma regra de três simples para o cálculo da distância. Esta equação pode ser vista na Equação 2, onde o valor para d é a nova distância a ser encontrada, o valor para d_0 é a distância medida a ser usada como base. Da mesma forma, o valor para w é a largura encontrada pelas leituras da câmera, e o valor para w_0 é a largura base usada como referência, salva no arquivo .json

$$d = \frac{d_0 * w_0}{w} \quad (2)$$

5.2 ALGORITMO PROPOSTO PARA O POSICIONAMENTO DOS MARCADORES

O conceito em torno do algoritmo de posicionamento dos marcadores é, após o reconhecimento da *label* pela câmera, e após a determinação de sua distância pelo método dos cálculos realizados na Equação 2, determinar a posição cartesiana do objeto correspondente da *label* no mapa.

Para isso, a primeira coisa que deve ser obtida para ter o resultado desse cálculo, é a distância da *label* para o posicionamento atual do robô, em metros. Essa distância pode ser chamada de *d*, assim como foi usada na Equação 2.

Pode-se utilizar o tópico *odometry* para adquirir a odometria do robô. O tópico relacionado é capaz de fornecer a posição atual do robô. Essa posição, porém, é dada em *quaternions*, possuindo um valor de *x*, *y*, *z* e *w*, valores que não apenas são relacionados à posição cartesiana do robô, mas também à sua orientação no plano bidimensional.

Para isso, o ROS possui transformadores específicos em sua biblioteca. O transformador *euler_from_quaternion* é capaz de fazer a conversão dos pontos em *quaternion* para ângulos de Euler. Os ângulos de Euler, representados por *roll*, *pitch* e *yaw*. Dentre esses, o mais adequado para representar a rotação do robô é o *yaw*, que pode ser obtido pela Equação 3, representado pelo símbolo grego theta, e onde todos os graus de *quaternion* são representados por *x*, *y*, *z* e *w*.

$$\theta = \arctan\left(\frac{2(zw + xy)}{1 - 2(y^2 + z^2)}\right) \quad (3)$$

Com o θ e o *d* em mãos, é possível fazer o cálculo da distância cartesiana entre o ponto onde a *label* foi encontrada e o ponto onde o robô está atualmente localizado. Essa distância delta pode ser encontrada nas Equação 4 e 5, servindo tanto para o eixo *x* quanto para o eixo *y*, onde a distância delta para cada eixo é representada por *dx* (Equação 4) e *dy* (Equação 5).

$$\Delta dx = d * \left(\cos \theta + \frac{\pi}{2}\right) \quad (4)$$

$$\Delta dy = d * \left(\sin \theta + \frac{\pi}{2}\right) \quad (5)$$

Com os valores encontrados, eles podem ser somados aos valores de x e y no plano cartesiano, localizando assim a posição cartesiana do *label* localizado pela câmera. Isso pode ser visto nas Equações 6 e 7.

$$x_{label} = x + \Delta dx \quad (6)$$

$$y_{label} = y + \Delta dy \quad (7)$$

5.3 RESULTADOS

Nessa seção, é feita a análise de resultados referentes tanto ao cálculo da distância quanto ao posicionamento dos marcadores, provando que o algoritmo descrito funciona da maneira esperada.

5.3.1 Do Cálculo da Distância

Para comprovar que o algoritmo de cálculo de distância está funcionando conforme o esperado, uma série de testes foi realizada no simulador. Para esses testes, a distância foi medida nos resultados fornecidos pelo algoritmo e na distância real fornecida pelo simulador, que corresponde ao valor real.

Com esses resultados em mãos, pôde-se criar uma tabela, apresentando os valores da distância calculada com o algoritmo, da distância real e do erro médio para cada um dos *labels*. Para o funcionamento correto do sistema, o objeto detectado precisava estar localizado no centro do quadro da câmera, para garantir que a distância medida seria em uma linha reta.

Os resultados, reunidos na Tabela 1, foram amplamente satisfatórios, com baixos erros médios e distâncias próximas às medidas fora do algoritmo. Desde que as medidas de referência fossem adequadas, o algoritmo teve pouco ou nenhum problema em identificar uma distância precisa.

Também foram realizados testes com o algoritmo fora do ambiente simulado, em uma situação real, com o robô AMPharmy físico. Nessa situação, foram detectadas as distâncias medidas de *banners* e cadeiras localizadas dentro do espaço do Habitat Senai. Os resultados desses testes foram até melhores que os simulados, com uma margem de erro abaixo de cinco por cento, dentro das imagens detectadas pela plataforma da YOLOv8 no modelo de *Transfer*

	Distância Calculada	Distância Real	Média do Erro
small stretcher	2.59	2.70	4%
wheelchair	2.34	2.50	6%
elevator	5.12	5.00	2%
large stretcher	2.82	2.32	17%

Tabela 1 – Comparação das Distâncias Calculadas e Reais com Média do Erro entre elas

Learning.

5.3.2 Posicionamento dos Marcadores

Para comprovar também que o posicionamento dos marcadores possui resultados satisfatórios, uma estratégia semelhante à anterior foi utilizada, com uma bateria de testes sendo realizada no simulador, tentando verificar o posicionamento do marcador proposto em comparação com o marcador que deveria estar no lugar real.

Com isso, o que se utilizou foram *labels* de teste, onde o robô deveria reconhecer a *label*, calcular sua posição e colocar o marcador no local ideal. Na Figura 21 pode-se ver o teste relacionado à *label Wheelchair*, onde se testou a posição relativa do marcador em relação à posição inicial. A cadeira de rodas do ambiente simulado se encontrava na frente da porta mais à direita do robô, portanto, mostrando um sucesso do marcador ao identificar a posição geral do objeto visto pela câmera.

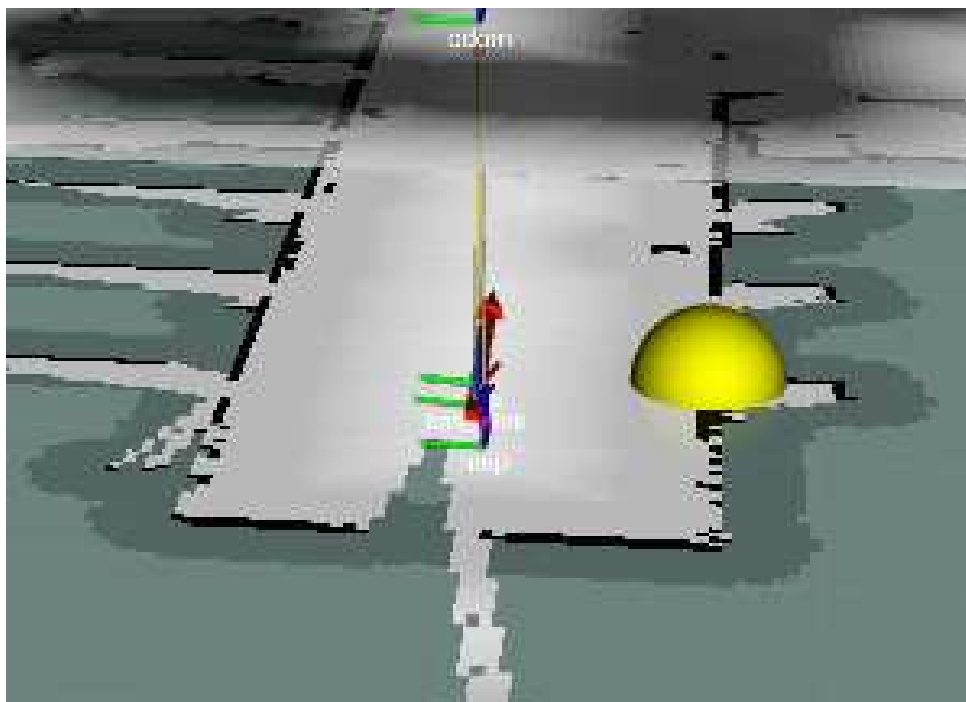


Figura 21 – O marcador da *label Wheelchair*, visto no RViz.

Fonte: Autor.

	Posição do Marcador	Posição do Objeto	Erro
ângulo de 65°	[0,699 -0,651]	[1,155 -0,530]	[0,456 0,121]
ângulo de 45°	[1,075 -0,909]	[1,155 -0,530]	[0,080 0,379]
ângulo de 90°	[0,976 -0,944]	[1,155 -0,530]	[0,179 0,414]

Tabela 2 – Comparação dos Marcadores com a Posição Real do Objeto

Após isso, esses marcadores tiveram suas posições comparadas com a posição definitiva do objeto no mapa. Foram realizados testes de comparação do posicionamento dessas duas posições dentro do plano do mapa simulado para verificar a acurácia do modelo, também buscando verificar não apenas o posicionamento do marcador, mas o quanto ele reconhece o objeto em angulações diferentes da câmera. É importante perceber que no cenário em questão, o objeto está completamente posicionado na frente do robô é considerado como um ângulo de noventa graus.

Assim sendo, se pode ver o resultado em uma tabela que mostra uma comparação das métricas analisadas com o resultado real, comparando a posição do marcador com onde o objeto realmente estava no mapa virtual no modelo cartográfico. Isso pode ser visto na representação da Tabela 2. Pode-se analisar que os resultados são extremamente satisfatórios nesse cenário, onde nenhum erro no eixo x ou y acaba por ser maior que meio metro. Em um cenário específico, o erro no eixo x acaba por chegar a uma faixa quase que irrelevante, de oito centímetros. Não apenas isso, mas o modelo, que foi majoritariamente projetado para a detecção de distâncias em linha reta, isso é, na angulação de noventa graus, também apresentou valores satisfatórios com ângulos mais orientados para o lado.

Sendo assim, a ferramenta prova seu funcionamento adequado, criando os marcadores em posições relativamente similares às esperadas. Esse posicionamento dos marcadores em composição com a rotina de detecção de imagens é capaz de criar um algoritmo que auxilie composições de navegação autônoma, especialmente com o objetivo em mente de criar uma navegação aperfeiçoada para a sua utilização dentro de hospitais, entregando medicamentos e auxiliando a equipe médica.

O autor dessa dissertação também conseguiu com sucesso publicar e apresentar o estudo referente à essa parte de identificação de distância via câmera no Simpósio Latino-Americano de Robótica (LARS) de 2023, cujo artigo pode ser visto em (FABRICIO *et al.*, 2023b).

6 CONCLUSÕES E PERSPECTIVAS

Este artigo propõe o desenvolvimento de um robô móvel em um ambiente hospitalar, visando automatizar a entrega de medicamentos e objetos diversos dentro do edifício, com o objetivo não apenas de aumentar a agilidade e a execução de processos dentro do hospital, mas também de ajudar os médicos a permanecerem por períodos mais longos ao lado de seus pacientes. Além disso, pode auxiliar em tornar o transporte de medicamentos ao redor do hospital uma tarefa mais segura.

Os resultados gerais deste projeto demonstram ser bastante satisfatórios, com o modelo de visão computacional funcionando conforme o esperado e fornecendo resultados geralmente excelentes, como pode ser visto nas curvas que representam o modelo. As marcações estão operando conforme o previsto, com o algoritmo responsável pelo cálculo da posição das etiquetas fornecendo resultados precisos.

Para projetos futuros, os próximos passos são levar este projeto para o robô móvel real, fazendo com que o AMPharmy trabalhe em um ambiente hospitalar e valide o sistema, entregando medicamentos e ajudando a equipe de trabalho conforme necessário. Um aumento no uso de sensores também pode ser útil, com a adição de um segundo LiDAR na parte traseira para dar ao robô mais liberdade de manobra, ou sensores IMU. Dada a configuração não ideal dos testes reais, uma amostra maior de imagens também precisa ser capturada e processada para um modelo mais preciso. Além disso, o projeto pode ser expandido além da entrega de medicamentos à medida que o modelo se torna mais refinado em interações futuras.

REFERÊNCIAS

ALBAWI, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. Understanding of a convolutional neural network. *In: IEEE. 2017 international conference on engineering and technology (ICET)*. [S.l.], 2017. p. 1–6.

ALKENDI, Yusra; SENEVIRATNE, Lakmal; ZWEIRI, Yahya. State of the art in vision-based localization techniques for autonomous navigation systems. **IEEE Access**, IEEE, v. 9, p. 76847–76874, 2021.

ARKIN, Ronald C; MURPHY, Robin R. Autonomous navigation in a manufacturing environment. **IEEE transactions on robotics and automation**, v. 6, n. 4, p. 445–454, 1990.

BELLEMARE, Marc G; CANDIDO, Salvatore; CASTRO, Pablo Samuel; GONG, Jun; MACHADO, Marlos C; MOITRA, Subhdeep; PONDA, Sameera S; WANG, Ziyu. Autonomous navigation of stratospheric balloons using reinforcement learning. **Nature**, Nature Publishing Group UK London, v. 588, n. 7836, p. 77–82, 2020.

BHATTACHARYA, Sweta; MADDIKUNTA, Praveen Kumar Reddy; PHAM, Quoc-Viet; GADEKALLU, Thippa Reddy; CHOWDHARY, Chiranjil Lal; ALAZAB, Mamoun; PIRAN, Md Jalil *et al.* Deep learning and medical image processing for coronavirus (covid-19) pandemic: A survey. **Sustainable cities and society**, Elsevier, v. 65, p. 102589, 2021.

BHAVSAR, P. Transfer Learning In NLP. Available in <https://medium.com/modern-nlp/transfer-learning-in-nlp-f5035cc3f62f>. [S.l.: s.n.], 2019.

BRADSKI, Gary; KAEHLER, Adrian *et al.* Opencv. **Dr. Dobb's journal of software tools**, v. 3, n. 2, 2000.

BRETTHAUER, David. Open source software: A history. 2001.

BUCKLAND, Michael; GEY, Fredric. The relationship between recall and precision. **Journal of the American society for information science**, Wiley Online Library, v. 45, n. 1, p. 12–19, 1994.

BUDD, Samuel; ROBINSON, Emma C; KAINZ, Bernhard. A survey on active learning and human-in-the-loop deep learning for medical image analysis. **Medical Image Analysis**, Elsevier, v. 71, p. 102062, 2021.

BURGER, Benjamin; MAFFETTONE, Phillip M; GUSEV, Vladimir V; AITCHISON, Catherine M; BAI, Yang; WANG, Xiaoyan; LI, Xiaobo; ALSTON, Ben M; LI, Buyi; CLOWES,

Rob *et al.* A mobile robotic chemist. **Nature**, Nature Publishing Group UK London, v. 583, n. 7815, p. 237–241, 2020.

CALKINS, Frederick T; FLATAU, Alison B; DAPINO, Marcelo J. Overview of magnetostrictive sensor technology. **Journal of Intelligent Material Systems and Structures**, Sage Publications Sage UK: London, England, v. 18, n. 10, p. 1057–1066, 2007.

CARDONA, Manuel; CORTEZ, Fernando; PALACIOS, Andrés; CERROS, Kevin. Mobile robots application against covid-19 pandemic. *In: IEEE. 2020 IEEE ANDESCON. [S.l.]*, 2020. p. 1–5.

CASTRO, Jose Dieguez. **Introducing Linux Distros. [S.l.]**: Springer, 2016.

CIVITA, Andrea; FIORI, Simone; ROMANI, Giuseppe. A mobile acquisition system and a method for hips sway fluency assessment. **Information**, MDPI, v. 9, n. 12, p. 321, 2018.

COELHO, Averaldo Alencar; SOUZA, Mikael Moraes de; SANTOS, Leon Denis Rodrigues dos. A importância da nr-12 segurança no trabalho em máquinas e equipamentos. 2018.

DAZA, Iván García; RENTERO, Mónica; MALDONADO, Carlota Salinas; GONZALO, Ruben Izquierdo; PARRA, Noelia Hernández; BALLARDINI, Augusto; LLORCA, David Fernandez. Fail-aware lidar-based odometry for autonomous vehicles. **Sensors**, MDPI, v. 20, n. 15, p. 4097, 2020.

DWYER, B.; NELSON, J. (2022), solawetz, j. **et. al. Roboflow (Version, //roboflow.com. computer vision, [Software]. Available from https, v. 1., 2022.**

EVANS, John M. Helpmate: An autonomous mobile robot courier for hospitals. *In: IEEE. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94). [S.l.]*, 1994. v. 3, p. 1695–1700.

FABRICIO, Felipe; OLIVEIRA, André Schneider de; TEIXEIRA, AS Marco. Indoor autonomous robot: Perception of the environment through computer vision. *In: IEEE. 2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE). [S.l.]*, 2023. p. 543–548.

FABRICIO, Felipe; OLIVEIRA, André Schneider de; TEIXEIRA, Marco AS; GÜNTHER, Gustavo. Autonomous mobile pharmacy (ampharmacy): Identification of hospital objects and their respective positions. *In: IEEE. 2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE). [S.l.]*, 2023. p. 532–536.

FEDERAL, Polícia. **Polícia Civil apura desvio de medicamentos de hospitais e unidades de saúde de SP**. 2023. Disponível em: <https://www.gov.br/pf/pt-br/assuntos/noticias/2023/07/pf-cumpre-mandados-para-apurar-de-desvio-de-medicamentos>.

FELBO, Bjarke; MISLOVE, Alan; SØGAARD, Anders; RAHWAN, Iyad; LEHMANN, Sune. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. **arXiv preprint arXiv:1708.00524**, 2017.

GALVÃO, César. **Polícia Civil apura desvio de medicamentos de hospitais e unidades de saúde de SP**. 2020. Disponível em: <https://g1.globo.com/sp/sao-paulo/noticia/2020/08/27/policia-civil-apura-desvio-de-medicamentos-de-hospitais-e-unidades-de-saude-de-sp.ghhtml>.

GERMAN, Daniel M; GONZÁLEZ-BARAHONA, Jesús M. An empirical study of the reuse of software licensed under the gnu general public license. *In*: SPRINGER. **IFIP International Conference on Open Source Systems**. [S.l.], 2009. p. 185–198.

GOUTTE, Cyril; GAUSSIER, Eric. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. *In*: SPRINGER. **European conference on information retrieval**. [S.l.], 2005. p. 345–359.

GRUBER, Wolfgang; BÄCK, Wolfgang; AMRHEIN, Wolfgang. Design and implementation of a wheel hub motor for an electric scooter. *In*: IEEE. **2011 IEEE Vehicle Power and Propulsion Conference**. [S.l.], 2011. p. 1–6.

GUAN, Hao; LIU, Mingxia. Domain adaptation for medical image analysis: a survey. **IEEE Transactions on Biomedical Engineering**, IEEE, v. 69, n. 3, p. 1173–1185, 2021.

GUERREIRO, João; SATO, Daisuke; ASAKAWA, Saki; DONG, Huixu; KITANI, Kris M; ASAKAWA, Chieko. Cabot: Designing and evaluating an autonomous navigation robot for blind people. *In*: **Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility**. [S.l.: s.n.], 2019. p. 68–82.

HOLLAND, Jane; KINGSTON, Liz; MCCARTHY, Conor; ARMSTRONG, Eddie; O'DWYER, Peter; MERZ, Fionn; MCCONNELL, Mark. Service robots in the healthcare sector. **Robotics**, MDPI, v. 10, n. 1, p. 47, 2021.

HUNG, Lillian; GREGORIO, Mario; MANN, Jim; WALLSWORTH, Christine; HORNE, Neil; BERNDT, Annette; LIU, Cindy; WOLDUM, Evan; AU-YEUNG, Andy; CHAUDHURY, Habib. Exploring the perceptions of people with dementia about the social robot paro in a hospital setting. **Dementia**, SAGE Publications Sage UK: London, England, v. 20, n. 2, p. 485–504, 2021.

IQBAL, Jamshed; KHAN, Zeashan Hameed; KHALID, Azfar. Prospects of robotics in food industry. **Food Science and Technology**, SciELO Brasil, v. 37, p. 159–165, 2017.

JAVOID, Mohd; HALEEM, Abid; SINGH, Ravi Pratap; SUMAN, Rajiv. Substantial capabilities of robotics in enhancing industry 4.0 implementation. **Cognitive Robotics**, Elsevier, v. 1, p. 58–75, 2021.

JIANG, Saike; WANG, Shilin; YI, Zhongyi; ZHANG, Meina; LV, Xiaolan. Autonomous navigation system of greenhouse mobile robot based on 3d lidar and 2d lidar slam. **Frontiers in Plant Science**, Frontiers, v. 13, p. 815218, 2022.

JOO, Sung-Hyeon; MANZOOR, Sumaira; ROCHA, Yuri Goncalves; BAE, Sang-Hyeon; LEE, Kwang-Hee; KUC, Tae-Yong; KIM, Minsung. Autonomous navigation framework for intelligent robots based on a semantic environment modeling. **Applied Sciences**, MDPI, v. 10, n. 9, p. 3219, 2020.

JOSHI, Rashika; TRIPATHI, Meenakshi; KUMAR, Amit; GAUR, Manoj Singh. Object recognition and classification system for visually impaired. *In: IEEE. 2020 International Conference on Communication and Signal Processing (ICCSP)*. [S.l.], 2020. p. 1568–1572.

KAM, Hyeong Ryeol; LEE, Sung-Ho; PARK, Taejung; KIM, Chang-Hun. Rviz: a toolkit for real domain data visualization. **Telecommunication Systems**, Springer, v. 60, p. 337–345, 2015.

KARPLUS, Robert; LUTTINGER, JM. Hall effect in ferromagnetics. **Physical Review**, APS, v. 95, n. 5, p. 1154, 1954.

KRAFFT, Martin F. **The Debian system: concepts and techniques**. [S.l.]: No Starch Press, 2005.

KUZE, Naohiro; SHIBASAKI, Ichiro. Mbe research and production of hall sensors. **III-Vs Review**, Elsevier, v. 10, n. 1, p. 28–32, 1997.

KYRARINI, Maria; LYGERAKIS, Fotios; RAJAVENKATANARAYANAN, Akilesh; SEVASTOPOULOS, Christos; NAMBIAPPAN, Harish Ram; CHAITANYA, Kodur Krishna; BABU, Ashwin Ramesh; MATHEW, Joanne; MAKEDON, Fillia. A survey of robots in healthcare. **Technologies**, MDPI, v. 9, n. 1, p. 8, 2021.

LAPPALAINEN, Inka. Logistics robots as an enabler of hospital service system renewal. *In: The 10 years Naples forum on service. Service dominant logic, network and systems theory and service science: integrating three perspectives for a new service agenda*. Ischia, Italy. [S.l.: s.n.], 2019.

LI, You; IBANEZ-GUZMAN, Javier. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. **IEEE Signal Processing Magazine**, IEEE, v. 37, n. 4, p. 50–61, 2020.

MACENSKI, Steven; FOOTE, Tully; GERKEY, Brian; LALANCETTE, Chris; WOODALL, William. Robot operating system 2: Design, architecture, and uses in the wild. **Science Robotics**, v. 7, n. 66, p. eabm6074, 2022. Disponível em: <https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>.

MARMAGLIO, Paolo; CONSOLATI, Davide; AMICI, Cinzia; TIBONI, Monica. Autonomous vehicles for healthcare applications: A review on mobile robotic systems and drones in hospital and clinical environments. **Electronics**, MDPI, v. 12, n. 23, p. 4791, 2023.

MELOTTI, Gledson; PREMEBIDA, Cristiano; GONÇALVES, Nuno. Multimodal deep-learning for object recognition combining camera and lidar data. *In: IEEE. 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). [S.l.]*, 2020. p. 177–182.

NAVYA, Pulimi; RANJITH, R. Analysis of path planning algorithms for service robots in hospital environment. *In: IEEE. 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). [S.l.]*, 2021. p. 1–6.

NGUGI, Lawrence C; ABELWAHAB, Moataz; ABO-ZAHHAD, Mohammed. Recent advances in image processing techniques for automated leaf pest and disease recognition—a review. **Information processing in agriculture**, Elsevier, v. 8, n. 1, p. 27–51, 2021.

OHASHI, Fusao; KAMINISHI, Kohei; HEREDIA, Jorge David Figueroa; KATO, Hiroki; OGATA, Taiki; HARA, Tatsunori; OTA, Jun. Realization of heavy object transportation by mobile robots using handcarts and outrigger. **Robomech journal**, Springer, v. 3, p. 1–12, 2016.

PALIX, Nicolas; THOMAS, Gaël; SAHA, Suman; CALVÈS, Christophe; LAWALL, Julia; MULLER, Gilles. Faults in linux: Ten years later. *In: Proceedings of the sixteenth international conference on Architectural support for programming languages and operating systems. [S.l.: s.n.]*, 2011. p. 305–318.

PEREZ, Luis; WANG, Jason. The effectiveness of data augmentation in image classification using deep learning. arXiv, 2017.

QUIGLEY, Morgan; GERKEY, Brian; SMART, William D. **Programming Robots with ROS: a practical introduction to the Robot Operating System. [S.l.]: "O'Reilly Media, Inc."**, 2015.

RAJ, Thinal; HASHIM, Fazida Hanim; HUDDIN, Aqilah Baseri; IBRAHIM, Mohd Faisal; HUSSAIN, Aini. A survey on lidar scanning mechanisms. **Electronics**, MDPI, v. 9, n. 5, p. 741, 2020.

ROHMER, Eric; SINGH, Surya PN; FREESE, Marc. V-rep: A versatile and scalable robot simulation framework. *In: IEEE. 2013 IEEE/RSJ international conference on intelligent robots and systems. [S.l.]*, 2013. p. 1321–1326.

RUDER, Sebastian. An overview of gradient descent optimization algorithms. **arXiv preprint arXiv:1609.04747**, 2016.

SALVI, Massimo; ACHARYA, U Rajendra; MOLINARI, Filippo; MEIBURGER, Kristen M. The impact of pre-and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis. **Computers in Biology and Medicine**, Elsevier, v. 128, p. 104129, 2021.

SCHAHAWI, Magda Diab-El; ZINGG, Walter; VOS, Margreet; HUMPHREYS, Hilary; LOPEZ-CERERO, Lorena; FUESZL, Astrid; ZAHAR, Jean Ralph; PRESTERL, Elisabeth; GROUP”, ESCMID Study Group on Nosocomial Infections “The decontamination research working. Ultraviolet disinfection robots to improve hospital cleaning: Real promise or just a gimmick? **Antimicrobial Resistance & Infection Control**, Springer, v. 10, p. 1–3, 2021.

SCHALKWYK, DJ Van; KAMPER, MJ. Effect of hub motor mass on stability and comfort of electric vehicles. *In: IEEE. 2006 IEEE vehicle power and propulsion conference. [S.l.]*, 2006. p. 1–6.

SHIEH, Ming-Yuan; HSIEH, JC; CHENG, CP. Design of an intelligent hospital service robot and its applications. *In: IEEE. 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583). [S.l.]*, 2004. v. 5, p. 4377–4382.

SHIT, Rathin Chandra. Precise localization for achieving next-generation autonomous navigation: State-of-the-art, taxonomy and future prospects. **Computer Communications**, Elsevier, v. 160, p. 351–374, 2020.

SLABAUGH, Gregory G. Computing euler angles from a rotation matrix. **Retrieved on August**, v. 6, n. 2000, p. 39–63, 1999.

SUN, Jian; ZHAO, Jie; HU, Xiaoyang; GAO, Hongwei; YU, Jiahui. Autonomous navigation system of indoor mobile robots using 2d lidar. **Mathematics**, MDPI, v. 11, n. 6, p. 1455, 2023.

TANG, Yunchao; CHEN, Mingyou; WANG, Chenglin; LUO, Lufeng; LI, Jinhui; LIAN, Guoping; ZOU, Xiangjun. Recognition and localization methods for vision-based fruit picking robots: A review. **Frontiers in Plant Science**, Frontiers Media SA, v. 11, p. 510, 2020.

TAYLOR, Angelique; LEE, Hee Rin; KUBOTA, Alyssa; RIEK, Laurel D. Coordinating clinical teams: Using robots to empower nurses to stop the line. **Proceedings of the ACM on Human-Computer Interaction**, ACM New York, NY, USA, v. 3, n. CSCW, p. 1–30, 2019.

TORRIERI, Don. **Principles of spread-spectrum communication systems**. [S.l.]: Springer, 2005. v. 1.

VISA, Sofia; RAMSAY, Brian; RALESCU, Anca L; KNAAP, Esther Van Der. Confusion matrix-based feature selection. **Maics**, v. 710, n. 1, p. 120–127, 2011.

WANDALKAR, Poorwa; PANDIT, PT; ZITE, AR. Abc and ved analysis of the drug store of a tertiary care teaching hospital. **Indian Journal of Basic and Applied Medical Research**, v. 3, n. 1, p. 126–131, 2013.

WANG, Chien-Yao; BOCHKOVSKIY, Alexey; LIAO, Hong-Yuan Mark. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. **arXiv preprint arXiv:2207.02696**, 2022.

WEBSTER, Craig; IVANOV, Stanislav. **Robotics, artificial intelligence, and the evolving nature of work**. [S.l.]: Springer, 2020.

WIJAYASINGHE, Indika; RUTHS, Justin; BÜTTNER, Ulrich; GHOSH, Bijoy K; GLASAUER, Stefan; KREMMYDA, Olympia; LI, Jr-Shin. Potential and optimal control of human head movement using tait–bryan parametrization. **Automatica**, Elsevier, v. 50, n. 2, p. 519–529, 2014.

XIAO, Kai; ENGSTROM, Logan; ILYAS, Andrew; MADRY, Aleksander. **Noise or Signal: The Role of Image Backgrounds in Object Recognition**. 2020.

YANG, Guang-Zhong; CAMBIAS, James; CLEARY, Kevin; DAIMLER, Eric; DRAKE, James; DUPONT, Pierre E; HATA, Nobuhiko; KAZANZIDES, Peter; MARTEL, Sylvain; PATEL, Rajni V *et al.* **Medical robotics—Regulatory, ethical, and legal considerations for increasing levels of autonomy**. [S.l.]: American Association for the Advancement of Science, 2017. eaam8638 p.

YANG, Kaiyu; QINAMI, Klint; FEI-FEI, Li; DENG, Jia; RUSSAKOVSKY, Olga. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. *In: Proceedings of the 2020 conference on fairness, accountability, and transparency*. [S.l.: s.n.], 2020. p. 547–558.

YEE, Phang Darren Ren; PINRATH, Nattawat; MATSUHIRA, Nobuto. Autonomous mobile robot navigation using 2d lidar and inclined laser rangefinder to avoid a lower object. *In: IEEE. 2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. [S.l.], 2020. p. 1404–1409.

ZAMAN, Shahzad; COMBA, Lorenzo; BIGLIA, Alessandro; AIMONINO, Davide Ricauda; BARGE, Paolo; GAY, Paolo. Cost-effective visual odometry system for vehicle motion control in agricultural environments. **Computers and Electronics in Agriculture**, Elsevier, v. 162, p. 82–94, 2019.

ZHAO, Jingchao; GAO, Junyao; ZHAO, Fangzhou; LIU, Yi. A search-and-rescue robot system for remotely sensing the underground coal mine environment. **Sensors**, MDPI, v. 17, n. 10, p. 2426, 2017.

ZHONG, Yuanyi; WANG, Jianfeng; PENG, Jian; ZHANG, Lei. Anchor box optimization for object detection. *In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. [S.l.: s.n.], 2020. p. 1286–1294.

ZOU, Qin; SUN, Qin; CHEN, Long; NIE, Bu; LI, Qingquan. A comparative analysis of lidar slam-based indoor navigation for autonomous vehicles. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 23, n. 7, p. 6907–6921, 2021.

APÊNDICE A – APÊNDICE

A.1 ÂNGULOS DE EULER E TAIT-BRYAN

Ângulos de Euler são medidas essenciais para descrever a orientação de qualquer corpo no espaço, extremamente usados na engenharia, especialmente em corpos livres no ambiente (como foguetes) ou simulações, vistas muito em robótica, aviação ou desenvolvimento de jogos e animações.

Para isso, precisamos entrar na teoria de transformação de coordenadas (SLABAUGH, 1999). Tem-se que cada um dos ângulos são parâmetros independentes e arbitrários. Os ângulos referenciados não são exatamente considerados ângulos de Euler, mas uma subcategoria deles, chamados de ângulos de Tait-Bryan (WIJAYASINGHE *et al.*, 2014), ou Eixos de Aeronaves. A maior diferença deles para com ângulos de Euler propriamente ditos são que seu eixo não é fixo, sendo que ele é orientado com relação à sua primeira medida.

Esses ângulos, conhecidos como *roll*, *pitch* e *yaw*, são responsáveis pelo controle rotacional desses corpos livres no espaço, como pode ser visto na Figura 22. Roll pode ser considerado a rotação em torno do eixo transversal, Pitch a rotação em torno do eixo longitudinal e Yaw a rotação em torno do eixo vertical.

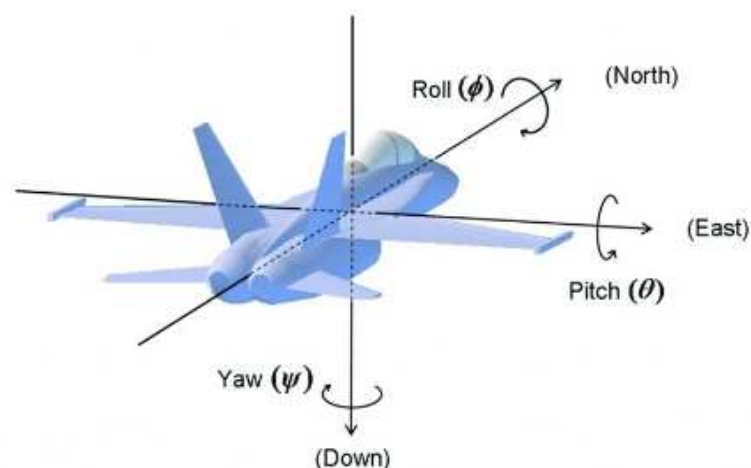


Figura 22 – Representação gráfica dos ângulos de Roll, Pitch e Yaw em uma aeronave.

Fonte: (CIVITA *et al.*, 2018).

Pode-se ver na Figura 22 que, numa representação real na figura de uma aeronave, *roll* indicaria o balanço, o manobrar do avião; *pitch* indicaria a inclinação, necessária para ganho e perda de altitude; *yaw* indicaria o giro do avião, mudando a direção de sua trajetória.

Para fins de robótica móvel no chão, o único que acaba sendo realmente relevante para esse projeto é o *yaw*, devido ao projeto não levantar voo, não precisando de inclinação ou manobra aérea, apenas do giro no próprio eixo. O próprio ROS já possui informações de como regular esses ângulos no tópico de velocidade *cmd_vel*. Nele, a mensagem incluída no tópico é separada em valores lineares e angulares de *x*, *y* e *z*, onde um robô móvel terrestre utiliza apenas o valor linear de *x* (visto que o robô não se desloca transversalmente) e o valor angular de *z* (o que representa, de fato, o *yaw*).

A.2 FIRMWARE

Um número considerável de *firmwares* diferentes foram usados para a construção do projeto descrito nesse trabalho. Nessa seção, esses programas serão explicados em mais detalhe, considerando a proposta desse trabalho: a construção do mapeamento semântico de um robô de navegação autônoma em um ambiente simulado hospitalar.

A.2.1 ROS

Criado em 2007 e desenvolvido pela Open Robotics, o ROS (*Robot Operational System*) é, na atualidade, a plataforma mais utilizada mundialmente para a execução de projetos robóticos, sendo parte integral de projetos em até mesmo na Sony, Boston Dynamics e NASA.

O ROS, embora tenha "sistema operacional" em seu nome, funciona como um conjunto de frameworks: diversas aplicações voltadas à robótica, unidas em apenas um ecossistema. Seu funcionamento se dá pelo esquema pub/sub de envio de mensagens, que é possível de se ver graficamente na Figura 23.

A explicação do que acontece na Figura 23 se dá pela comunicação entre dois chamados nós. Cada nó é responsável por uma concentração de informações, comandos e funções, geralmente localizadas dentro de um *script*, seja ele na linguagem de C++ ou Python.

Esses nós se comunicam entre si por meio de tópicos. Cada tópico é responsável pela captação e envio de um tipo de mensagem. Esse tipo pode ser um tipo base, como uma mensagem de um tipo *string*, *integer* ou *float*, como pode ser uma mensagem de tipos generalizados do ROS. Por exemplo, uma mensagem do tipo *Odom* contém várias informações sobre a odometria geral do robô, como seu posicionamento espacial no mapa ou a velocidade atual. Uma mensagem do tipo *LaserScan* contém detalhes sobre cada leitura de um laser e suas distâncias calculadas,

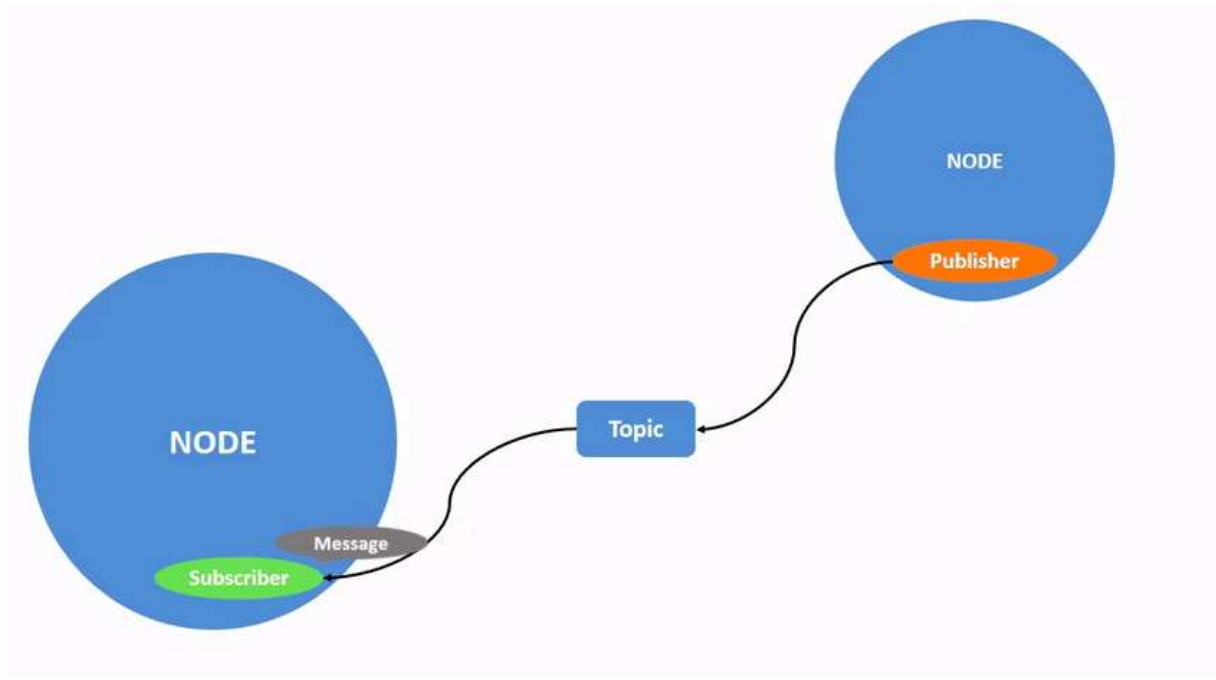


Figura 23 – Exemplo de envio de mensagens pelo ROS por um *publisher* e um *subscriber*.

Fonte: ROS2 Documentation (MACENSKI *et al.*, 2022).

além de campos indicando o nível de reflexividade. Não obstante, o usuário ainda pode criar seus próprios campos customizados.

A passagem de informações entre nós por meio de um tópico se dá por um *Publisher* e um *Subscriber*. É um estilo de troca de mensagens entre dois pontos (no caso, os dois nós), que é usado pelo ROS para envio e coleta de mensagens.

Um nó que possui um *Publisher* irá publicar a mensagem para um tópico, onde ela ficará por tempo indeterminado. Uma nova mensagem publicada no tópico irá sobrescrever a anterior, pois um tópico suporta apenas uma mensagem por vez. O *Subscriber*, porém, geralmente manipulado dentro do *script* por meio de *callbacks*, irá resgatar a mensagem atualmente localizada dentro do tópico, fazendo a manipulação dessa mensagem para uma variável dentro do nó, que pode ser utilizada como se bem entender.

Publishers e *Subscribers* podem ser usados das mais diversas maneiras. No caso de um robô autônomo, por exemplo, um nó que faz a leitura da velocidade das rodas pode constantemente publicá-la como mensagem para o tópico `/cmd_vel`, onde ela poderá ser resgatada por um outro nó que possui o propósito de projetar um controle da rota do aparelho por meio de seu status atual de velocidade.

O sistema também faz um uso constante de *launches*, arquivos com um formato específico XML projetados para executar vários nós de maneira simultânea, além de ser capaz de

definir parâmetros e até rodar outros *launches*.

O ROS é amplamente considerado como uma opção completa pra um projeto robótico devido à sua natureza *open-source*, por ser constantemente atualizado e por seu caráter comunitário de *frameworks*, onde novas funções especialmente bordadas para o universo da robótica são constantemente atualizadas e distribuídas pela comunidade do sistema.

A.2.2 Ubuntu 20.04

O Ubuntu é uma das distribuições mais conhecidas no cenário mundial de um sistema operacional baseado em Linux. O Linux em si pode ser descrito como um sistema operacional de código aberto baseado no kernel Linux (PALIX *et al.*, 2011). Desenvolvido por Linus Torvalds em 1991 (BRETTHAUER, 2001), esse kernel Linux é o núcleo do sistema operacional que interage diretamente com o hardware do computador. O sistema operacional completo, frequentemente referido como "GNU/Linux" devido à combinação do kernel Linux com as ferramentas do projeto GNU, é uma implementação completa de todo o sistema.

Uma característica importante do Linux é seu desenvolvimento voltado ao código aberto, significando que o código-fonte do sistema operacional está disponível para o público. Isso permite desenvolvedores por todo o globo contribuírem para o seu aprimoramento e personalização. Além disso, o Linux é distribuído sob os termos da Licença Pública Geral GNU (GPL) (GERMAN; GONZÁLEZ-BARAHONA, 2009), que garante a liberdade de executar, modificar e distribuir versões modificadas do software.

O Linux é também capaz de ser executado em uma variedade de plataformas, desde servidores até dispositivos embarcados e computadores pessoais. Graças à alta capacidade de modificações e portabilidade para diferentes ambientes, muitas distribuições do Linux estão disponíveis, cada uma delas adaptada para diferentes finalidades e preferências do usuário (CASTRO, 2016).

As distribuições Linux incluem uma variedade de utilitários, bibliotecas e aplicativos de software que formam um ambiente inteiro, completando-os como sistemas operacionais diversos. Exemplos de distribuições populares incluem Ubuntu, Fedora, Debian e CentOS. Cada uma dessas distribuições pode ter suas próprias interfaces gráficas, gerenciadores de pacotes e configurações específicas, mas todas compartilham o núcleo Linux como base, portanto, o conceito de que essas distribuições são todas baseadas em Linux.

O Ubuntu é uma das distribuições do Linux mais populares hoje presentes no mercado.

Ele é baseado no projeto Debian (KRAFFT, 2005), utilizando-se de sua arquitetura para toda a construção do sistema operacional. Suas primeiras versões oficiais datam do começo do século, e desde então novas versões, cada qual com mais upgrades e melhorias, são lançadas periodicamente, a aproximadamente cada seis meses. Atualmente, cada versão é nomeada com dois números. O primeiro indica o ano de lançamento, o segundo, o mês. Desde 2006, de forma bianual, no mês de abril, o Ubuntu lança sua versão LTS (Long-Term Support), que reúnem de forma confiável todas as atualizações recentes feitas para a plataforma e possuem suporte estendido de até cinco anos.

A versão do Ubuntu utilizada nesse projeto é a versão LTS mais recente compatível com ROS, o Ubuntu 20.04, de abril de 2020. apesar de já existirem versões mais novas, como a 23.10, essas versões não possuem uma versão mais atualizada do ROS, que teve seu desenvolvimento encerrado no lançamento do ROS Noetic Ninjemys. Uma *screenshot* do *Desktop* do Ubuntu 20.04 pode ser visto na Figura 24.

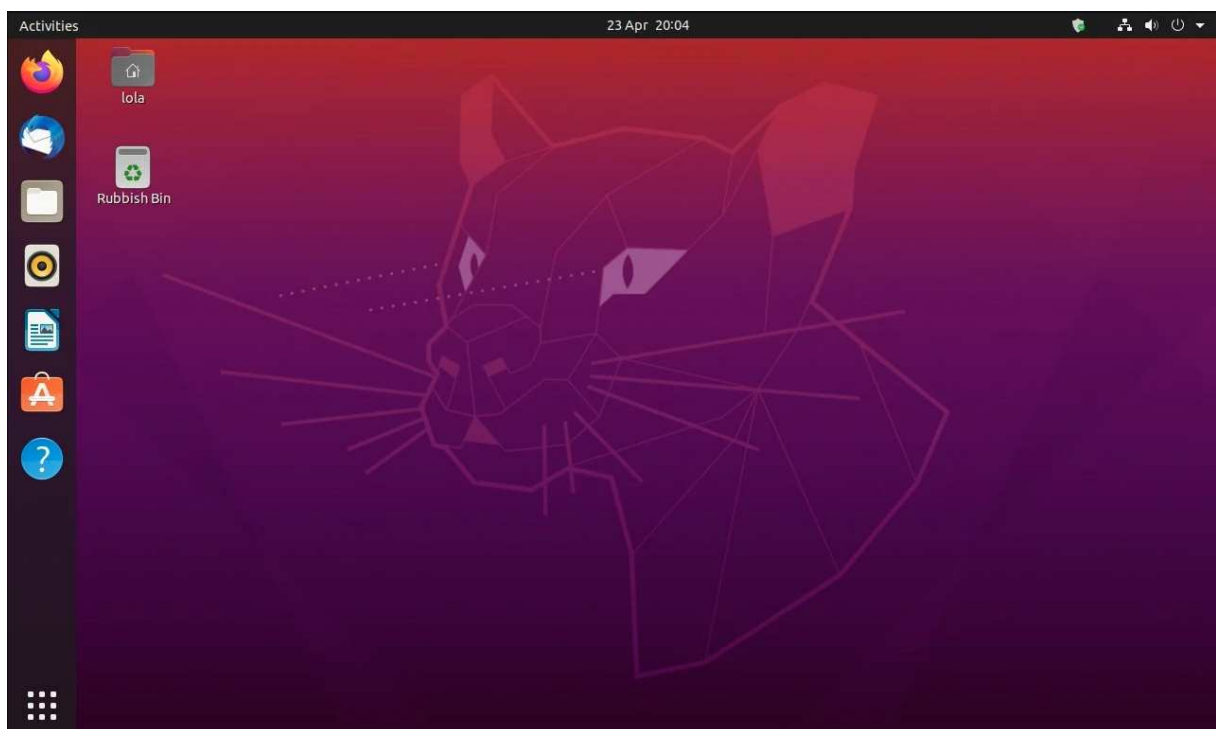


Figura 24 – Desktop do Ubuntu 20.04.

Fonte: Canonical Ltd.

A.2.3 CoppeliaSim

O CoppeliaSim, anteriormente conhecido por V-REP (*Virtual Robot Experimentation Platform*), é um *framework* de simulação de aplicações robóticas gerais. Desenvolvido pela

Coppelia Robotics AG, localizada em Zurique, na Suíça, o modelo é conhecido por ser uma das alternativas ao Gazebo, a alternativa nativa do ROS, porém mais robusta no fornecimento de diversas funções para simulação de uma forma onde a portabilidade nativa do sistema não seja prejudicada, habilitando uma pletera de simulações simultaneamente. (ROHMER *et al.*, 2013)

O simulador CoppeliaSim se baseia principalmente na existência de *scene objects*, que podem ser colocados juntos para a formação de um único objeto que junte todas as funções de seus antecessores, como, por exemplo, a junção de um LiDAR e uma câmera em um robô. O simulador possui suporte para objetos não apenas de formas básicas e junções, mas sensores de proximidade, visão e força. O CoppeliaSim também possui suporte à câmeras, luzes, objetos de referência (*dummies*), objetos complexos de volume, além de caminhos e gráficos.

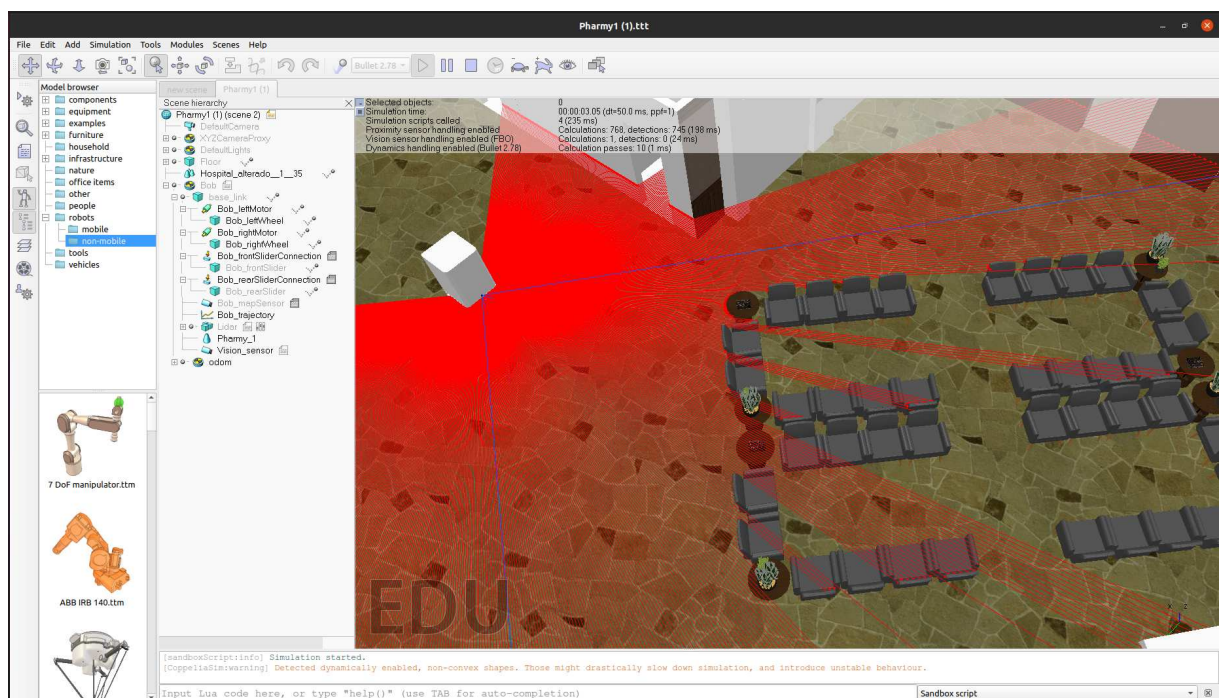


Figura 25 – CoppeliaSim em execução.

Fonte: Autor.

É possível ver na Figura 25, por exemplo, a execução de um modelo robótico navegando por um universo simulado, no qual o robô possui um LiDAR, visualizado pelas linhas vermelhas, e uma câmera, no qual seu campo de visão é determinado pelas linhas azuis. Um exemplo de como o robô simulado é construído pode ser visto também na barra lateral, onde se pode visualizar os modelos que constroem a parte física do robô e seus sensores todos integrados com *links*. A leitura desses sensores é feita por um código Lua, que publica todos os dados nos tópicos correspondentes do ROS.

A.2.4 RViz

O RViz (KAM *et al.*, 2015) é uma ferramenta de simulação robótica focada em simulações tridimensionais. A proposta de seu programa pode parecer similar ao que já se foi descrito no CoppeliaSim, porém, como escrito em (QUIGLEY *et al.*, 2015) em comparação com o Gazebo, alternativa ao Coppelia, a diferença entre os dois simuladores fica no fato em que o Gazebo mostra o que está acontecendo com o robô, de verdade, na simulação, enquanto o RViz mostra o que o robô imagina que está acontecendo com ele.

Isso se deve ao fato de que o RViz é uma plataforma potente para a leitura e visualização de sensores diversos. Enquanto o CoppeliaSim é capaz de simular o robô em todo o ambiente simulado, o RViz mostra a representação dessa simulação pelo sensoriamento interno ali realizado. Por exemplo, a leitura do LiDAR presente no robô simulado pode ser utilizada no RViz em uma conversão que transformará esses sinais em um mapa bidimensional em tempo real.

O aplicativo é capaz de simular várias das características necessárias para uma navegação autônoma de sucesso, como a visualização de mapas, sendo eles genéricos, locais, ou globais, de câmeras instaladas no robô, de *paths* a serem executados, também tanto globais quanto locais, entre outras. Isso se deve à aplicação de *widgets* presentes na plataforma, cada um representando uma visualização sensorial diferente. Esses *widgets* podem ser criados de antemão, podendo ter um *link* com um tópico correspondente após seu *setup*. Porém, também é possível obter uma lista de todos os tópicos em execução pelo RViz, que, verificando pelo tipo de mensagem, irá abrir uma árvore de sugestões para o usuário inserir a simulação diretamente no RViz por um tópico existente. Por exemplo, se o RViz detecta que há um tópico com uma mensagem do tipo LasersScan em execução, ele irá habilitar diretamente a construção de uma simulação de leitura de lasers já baseada naquele tópico para o usuário.

A.2.5 OpenCV

OpenCV (Open Source Computer Vision Library) (BRADSKI *et al.*, 2000) é uma biblioteca muito conhecida e difundida, especializada no desenvolvimento de funções que auxiliam na utilização de práticas de visão computacional. Desenvolvida primariamente pela Intel, teve como proposta majoritária desenvolver métodos que facilitassem o uso de qualquer aplicação voltada à visão computacional. Seu logotipo mostra um misto inteligente das letras C e V com a faixa de cores RGB.

Desde seu início, a biblioteca já evoluiu consideravelmente, abrangendo cada vez mais áreas (variando de reconhecimento facial, de movimentos, detecção de objetos, machine learning, entre outros), chegando até em 2011 a integrar o núcleo CUDA (Compute Unified Device Architecture), desenvolvido pela NVidia para suas placas gráficas, buscando acelerar o processamento de suas funções. O objetivo do OpenCV sempre foi fazer da visão computacional algo mais próximo e mais simples de ser manipulado pelo público em geral.

No âmbito desse projeto, o OpenCV foi amplamente utilizado para a detecção de imagens pela câmera, fazendo a manipulação e aquisição da imagem. Com ele, se é possível principalmente fazer a construção e posicionamento de *bounding boxes* para as leituras feitas na rede neural do sistema, além de que sua integração com Python permite simplificar em vários aspectos a leitura da imagem da câmera vinda diretamente de um tópico no ROS.

A.2.6 Ultralytics YOLOv8

O modelo da YOLOv8 é um modelo estado da arte que melhora em versões prévias de si mesmo, como a YOLOv7 (WANG *et al.*, 2022), que por si só já era cento e vinte vezes mais rápida que outra de suas versões anteriores, a YOLOv5. Ele utiliza um SSD (*Single-Shot Detection*) para a detecção de objetos, numa rede neural convolucional, a CNN. Essa construção de modelo faz ele perder na precisão de suas detecções, mas ganhar na velocidade, fazendo com que ela seja ideal para aplicações em tempo real, como o caso desse projeto.

O modelo funciona recebendo uma entrada, que, no caso, corresponde a uma imagem, e então passando por sua própria rede neural para gerar os resultados mais precisos da forma mais rápida possível. Essa rede de detecção é composta de vinte e quatro *layers* convolucionais, seguidos de dois *layers* finais completamente conectados. Desses, os vinte primeiros são pré-treinados pela utilização da ImageNet, uma *database* de imagens que é organizada para a hierarquia da WordNet, como pode ser explicado em (YANG *et al.*, 2020). Esses *layers* passam por um agrupamento generalizado no final, terminando o processo completamente conectados entre si.

Considerando que a adição de *layers* conectados no final para uma rede pré-treinada e convoluta geralmente melhora na sua performance, esse modelo pré-treinado e completamente conectado pela utilização da ImageNet é convertido para realizar a detecção de imagens. Após isso, o modelo começa a fazer a predição das imagens pela criação de várias *bounding boxes* por ela, dando as porcentagens nas quais o modelo tem a confiança de que aquela imagem

corresponde à uma das *labels* determinada, se valendo de modos como Non-Max Supression (NMS) para realizar a distinção das *bounding box* dentro de um objeto singular, removendo informações redundantes ou desnecessárias. Uma vez que esse valor chega numa porcentagem alta o suficiente, o *label* é confirmado, determinando a imagem com sua descrição de uma vez por todas.

A YOLOv8 em si possui algumas diferenças a mais do modelo descrito, que é padrão para todas as versões da YOLO. Uma de suas maiores novidades em versões recentes é a utilização de *anchor boxes*, que podem ser vistas de forma mais detalhada no artigo em (ZHONG *et al.*, 2020). Essas *anchor boxes* tentam pegar um objeto no qual elas se especializam mais na predição e tentam encontrar sua localização, tamanho e formato ideal, calculando o nível de intersecção entre as *bounding boxes*, chamado de Intersect Over Union. As *anchor boxes* também precisam ser corretamente parametrizadas para não acabarem grandes ou pequenas demais, o que acaba por ser um empecilho na detecção de objetos maiores e menores.

Um triunfo da estrutura YOLO também é a utilização das Efficient Layer Aggregation Networks (ELANs). É uma arquitetura eficiente, conseguindo um bom balanço de parâmetros, quantidade de cálculo e processamento computacional do modelo, conseguindo resultados de maneira rápida e eficiente. A YOLOv8 dá um passo além com esse modelo com a criação de Extended ELANs, as E-ELANs. Se valendo dos cálculos de *gradient descents*, algoritmos de otimização que arrumam os valores e parâmetros baseados no erro, que podem ser vistos de maneira mais particular em (RUDER, 2016).

A E-ELAN não muda o caminho geral dessas *gradient descents*: o que ela faz é expandir, embaralhar e fundir a cardinalidade do modelo sem que haja dano nos *gradient descents* originais, mudando a arquitetura do modelo apenas computacionalmente em sua execução, e não de forma estrutural. Isso faz com que a arquitetura da E-ELAN acabe por ser um grande trunfo da YOLOv8, permitindo-a aprender características mais complexas e diversas do que se parecia estar estagnado em versões anteriores, melhorando sua estabilidade e resultados. Uma versão gráfica do modelo da E-ELAN, comparando sua estrutura não apenas com a ELAN, mas com outros modelos, como VoVNet e CSPVoVNet, pode ser visto na Figura 26.

Mais um triunfo da YOLOv8 é a introdução da plataforma da Ultralytics. O *Ultralytics Hub*, como é chamado, é uma plataforma *cloud-based* integrada com uma versão *web* e uma *mobile*, integração com a aplicação Roboflow, construção de um pacote Pip para simplificação da instalação, entre outras. Todo esse *hub* de informações possibilita uma plataforma que visa

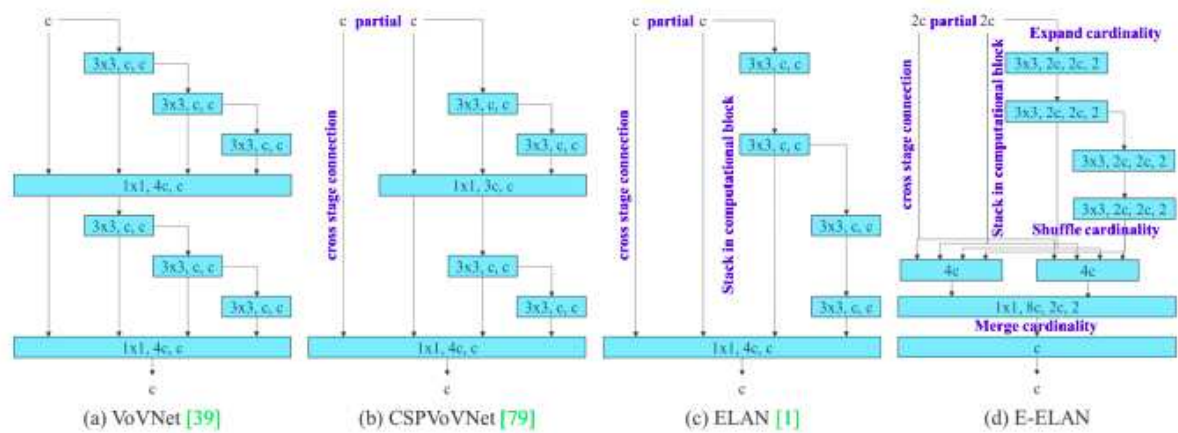


Figura 26 – A proposta de funcionamento da E-ELAN baseado em modelos mais antigos.

Fonte: Chien-Yao Wang (WANG *et al.*, 2022).

facilitar a construção de modelos de detecção de imagem, fazendo com que a construção de um *dataset* novo seja simplificada e feita pelo próprio aplicativo, podendo ser construída na plataforma *web* e rodar na plataforma *mobile* pela própria câmera do celular do usuário.