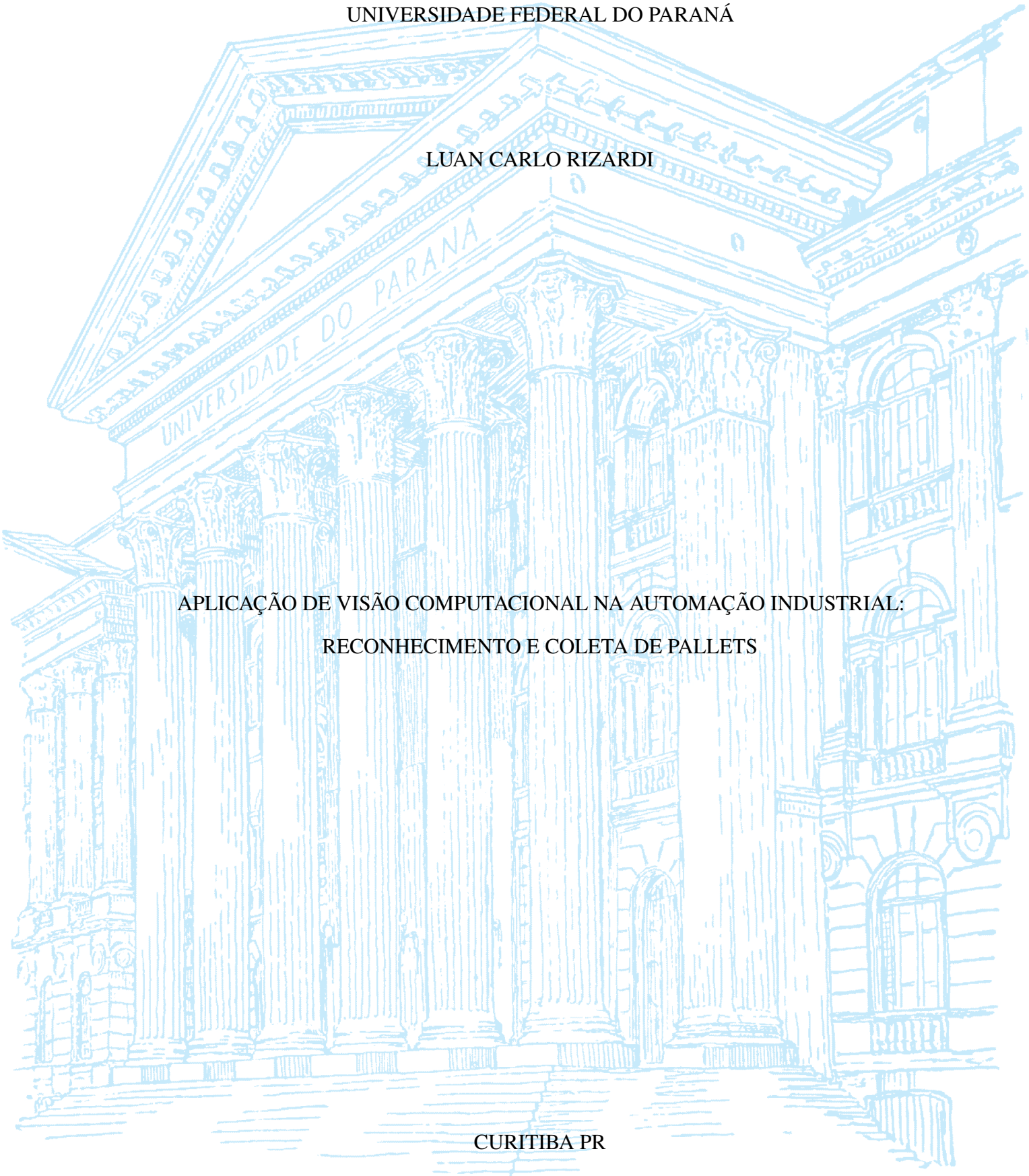
UNIVERSIDADE FEDERAL DO PARANÁ

LUAN CARLO RIZARDI



APLICAÇÃO DE VISÃO COMPUTACIONAL NA AUTOMAÇÃO INDUSTRIAL: RECONHECIMENTO E COLETA DE PALLETS

CURITIBA PR

2025

LUAN CARLO RIZARDI

APLICAÇÃO DE VISÃO COMPUTACIONAL NA AUTOMAÇÃO INDUSTRIAL: RECONHECIMENTO E COLETA DE PALLETS

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Computação*.

Orientador: Eduardo Todt.

CURITIBA PR

2025

"Entregue o seu caminho ao Senhor; confie nele, e ele agirá."— Salmo 37:5

AGRADECIMENTOS

Agradeço primeiramente a Deus, que foi minha força quando me faltou ânimo, minha luz quando o caminho parecia incerto e meu refúgio em todos os momentos de dúvida. Cada passo desta jornada só foi possível porque Ele me sustentou, me guiou e renovou minhas esperanças diariamente.

Agradeço, com todo o meu coração, à minha família e aos meus amigos. Mesmo distantes fisicamente, nunca deixaram de estar presentes de verdade. Suas palavras, orações, mensagens, incentivos e gestos de carinho foram abrigo nos dias difíceis e celebração nos dias de vitória. Vocês acreditaram em mim quando eu mesmo duvidei, e essa conquista também é de vocês.

Agradeço ainda aos professores que tive ao longo do curso, que com paciência, dedicação e compromisso não apenas ensinaram conteúdos, mas ajudaram a moldar minha trajetória. Cada orientação, cada desafio proposto e cada aprendizado compartilhado contribuiu profundamente para minha formação acadêmica e pessoal.

A todos que caminharam comigo, de perto ou de longe, deixo minha gratidão mais sincera. Este trabalho é fruto de fé, apoio e muito amor.

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de visão computacional embarcado que permite a um Veículo Guiado Automatizado (AGV) detectar e coletar pallets de forma autônoma, mesmo quando estes estão mal posicionados no ambiente industrial. O problema abordado surge de um cenário real de operação, no qual o desalinhamento frequente de pallets gera falhas na coleta e exige intervenções manuais, reduzindo a eficiência logística. A solução proposta integra segmentação de instâncias com YOLOv8n-seg, treinado com um dataset personalizado anotado no CVAT, e análise de profundidade obtida pela câmera 3D Lanxin MRDVS S2. A partir da máscara segmentada, extraem-se automaticamente três pontos relevantes (esquerda, centro e direita), posteriormente convertidos para coordenadas tridimensionais por meio do mapa de profundidade. Esses pontos são utilizados para estimar a pose completa do pallet, incluindo posição e orientação, que é transformada do referencial da câmera para o mapa global utilizando tf2_ros. De posse da pose corrigida, o sistema gera dinamicamente pontos de navegação alinhados ao pallet, permitindo que o AGV ajuste sua trajetória em tempo real e execute a aproximação e a coleta com precisão, mesmo em casos de desalinhamento significativo. Os experimentos realizados em ambiente industrial real demonstraram elevada precisão de segmentação (mAP@50 superior a 0,97), estabilidade na reconstrução da pose e execução bem-sucedida da coleta autônoma em diferentes condições. A abordagem proposta mostrou-se robusta, escalável e compatível com a arquitetura pré-existente do AGV, reduzindo intervenções humanas e aumentando a confiabilidade da operação logística. Palavras-chave: AGV. automação industrial. detecção de objetos. logística. segmentação de imagem. visão computacional.

Palavras-chave: AGV. automação industrial. detecção de objetos. logística. segmentação de imagem. visão computacional.

ABSTRACT

This work presents the development of an embedded computer vision system that enables an Automated Guided Vehicle (AGV) to autonomously detect and pick up pallets, even when they are misaligned in the industrial environment. The problem addressed originates from real operational conditions, where improperly positioned pallets frequently lead to pickup failures and require manual intervention, reducing logistical efficiency. The proposed solution integrates instance segmentation using YOLOv8n-seg, trained on a custom CVAT-annotated dataset, with depth analysis provided by the Lanxin MRDVS S2 3D camera. From each segmented mask, three key points (left, center, and right) are automatically extracted and converted into 3D coordinates using the depth map. These points are then used to estimate the pallet's full pose, including position and orientation, which is transformed from the camera frame to the global map via tf2_ros. With the corrected pose, the system dynamically generates navigation points aligned with the pallet, allowing the AGV to adapt its trajectory in real time and perform accurate approach and pickup, even under significant misalignment. Experiments conducted in a real industrial environment demonstrated high segmentation accuracy (mAP@50 above 0.97), stable pose reconstruction, and successful autonomous pallet pickup across multiple conditions. The proposed approach proved robust, scalable, and compatible with the AGV's existing architecture, reducing the need for human intervention and increasing the reliability of the logistic operation. Keywords: AGV. industrial automation. object detection. logistics. image segmentation. computer vision.

Keywords: AGV. industrial automation. object detection. logistics. image segmentation. computer vision.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

AGV	Veículo Guiado Automatizado, do inglês Automated Guided Vehicle
AMP	Precisão Mista Automática, do inglês Automatic Mixed Precision
AMR	Robô Móvel Autônomo, do inglês Autonomous Mobile Robot
CVAT	Ferramenta de Anotação de Visão Computacional, do inglês Computer Vision Annotation Tool
DINF	Departamento de Informática
ERP	Planejamento de Recursos Empresariais, do inglês Enterprise Resource Planning
IA	Inteligência Artificial
IoT	Internet das Coisas, do inglês Internet of Things
MES	Sistema de Execução de Manufatura, do inglês Manufacturing Execution System
PID	Controlador Proporcional–Integral–Derivativo, do inglês Proportional–Integral–Derivative
PCL	Biblioteca de Nuvem de Pontos, do inglês Point Cloud Library
PPGINF	Programa de Pós-Graduação em Informática
RGB-D	Imagem RGB com canal adicional de profundidade (Red–Green–Blue + Depth)
ROS	Sistema Operacional de Robôs, do inglês Robot Operating System
SLAM	Localização e Mapeamento Simultâneos, do inglês Simultaneous Localization and Mapping
UFPR	Universidade Federal do Paraná
VRAM	Memória de Vídeo, do inglês Video RAM
WMS	Sistema de Gerenciamento de Armazém, do inglês Warehouse Management System
YOLO	Arquitetura de detecção You Only Look Once

LISTA DE SÍMBOLOS

α	alfa, primeira letra do alfabeto grego
β	beta, segunda letra do alfabeto grego
γ	gama, terceira letra do alfabeto grego
ω	ômega, última letra do alfabeto grego
π	constante pi
τ	constante de tempo / tempo de resposta do sistema
θ	ângulo de orientação / ângulo de incidência
x	coordenada cartesiana no eixo horizontal
y	coordenada cartesiana no eixo vertical
d	distância do objeto à câmera

SUMÁRIO

1 INTRODUÇÃO

A automação da logística tem avançado significativamente com a adoção de robôs móveis autônomos, como *Automated Guided Vehicles* (AGVs), amplamente utilizados no transporte de pallets em centros de distribuição e fábricas (Campilho e Silva, 2023). Mesmo assim, muitos processos ainda dependem da colaboração humana, como a colocação dos pallets nas áreas de coleta. Quando essa tarefa não é executada corretamente, por exemplo, quando o pallet é posicionado fora da zona esperada, o AGV pode falhar na tentativa de coleta, exigindo correção manual e impactando negativamente a eficiência operacional (Ivanov et al., 2019).

Neste contexto, este trabalho propõe o desenvolvimento de um sistema de visão computacional embarcado no AGV, capaz de detectar e coletar pallets de forma autônoma, mesmo quando estes não estão perfeitamente posicionados. A proposta envolve o uso de técnicas de detecção de objetos YOLOv8 (Ultralytics, 2023) e análise de nuvem de pontos para estimar a posição tridimensional do pallet (Hu et al., 2020). As informações visuais serão integradas ao sistema de navegação do AGV, que se utilizará de um controlador Proporcional–Integral–Derivativo (PID) para alinhar-se ao pallet de forma precisa e eficiente, uma técnica já consolidada na robótica móvel (Åström e Murray, 2008);(Wang et al., 2017).

O projeto parte de um caso real observado no ambiente industrial da empresa onde este trabalho foi desenvolvido, e busca solucionar um problema recorrente de operação. Além da proposta técnica, este TCC aborda também a metodologia de definição do tema, fundamentada na revisão de literatura orientada por palavras-chave relevantes como AGV, *Autonomous Mobile Robot* (AMR), identificação e coleta de pallets, YOLOv8, sistema de visão, ambiente industrial, etc. Foram consultados e analisados diversos artigos científicos com foco em soluções similares, o que embasou e direcionou as decisões do projeto (Sapkota et al., 2024);(Golroudbari e Sabour, 2023).

Ao propor uma solução prática e aplicável ao ambiente industrial, este trabalho pretende contribuir com a evolução da navegação autônoma em robôs móveis e com a integração mais eficaz entre sistemas inteligentes de visão e controle em aplicações de logística 4.0 (Fragapane et al., 2022);(Schwab, 2017).

1.1 OBJETIVOS

O objetivo principal desse trabalho é desenvolver um sistema embarcado de visão computacional para AGVs, utilizando YOLOv8 e análise de nuvem de pontos, com o intuito de permitir a detecção e a coleta autônoma de pallets mesmo quando estes estão mal posicionados no ambiente industrial.

1.1.1 Objetivos Específicos

- Realizar uma revisão da literatura sobre técnicas de visão computacional aplicadas à identificação de pallets e à navegação de AGVs;
- Investigar e selecionar combinações de palavras-chave para levantamento de artigos relevantes na definição do escopo do projeto;
- Anotar um dataset de imagens industriais de pallets, utilizando ferramentas como CVAT, para treinar um modelo YOLOv8;

- Treinar e avaliar o desempenho de modelos de detecção de objetos e segmentação de instâncias para identificar pallets em diferentes posições;
- Integrar o sistema de detecção ao controle do AGV com um controlador PID para alinhamento e coleta do pallet;
- Testar o sistema em ambiente real ou simulado, validando sua eficácia em condições diversas de posicionamento dos pallets.

1.2 ORGANIZAÇÃO DO DOCUMENTO

Capítulo 2 — Fundamentação Teórica apresenta os conceitos essenciais que sustentam o trabalho, incluindo AGVs, visão computacional, câmeras 3D, construção de datasets e controle robótico.

Capítulo 3 — Metodologia descreve detalhadamente todas as etapas da solução proposta, desde a aquisição das imagens até a geração dos pontos corrigidos de navegação.

Capítulo 4 — Resultados e Discussão apresenta os resultados experimentais obtidos e discute seu significado à luz dos objetivos do trabalho.

Capítulo 5 — Conclusão sintetiza as principais contribuições do trabalho e apresenta suas perspectivas de continuidade.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 INDÚSTRIA 4.0

A Indústria 4.0 representa uma nova era da automação industrial, caracterizada pela integração entre sistemas físicos e digitais por meio de tecnologias como Internet das Coisas (IoT), Inteligência Artificial (IA), computação em nuvem e robótica avançada (Schwab, 2017; Campilho e Silva, 2023). Essa revolução visa não apenas aumentar a produtividade, mas também promover maior flexibilidade, personalização e eficiência energética nos processos industriais (Fragapane et al., 2022).

No contexto da logística, a Indústria 4.0 introduz conceitos como sistemas ciber-físicos e comunicação máquina-a-máquina, permitindo que robôs móveis autônomos, sensores inteligentes e plataformas analíticas colaborem em tempo real (Ivanov et al., 2019). Esse nível de conectividade permite operações adaptativas, com capacidade de autoajuste a falhas humanas ou variações no ambiente, como o mau posicionamento de pallets em áreas de coleta.

Pesquisas recentes destacam ainda o papel da robótica inteligente como elemento-chave da nova geração de fábricas conectadas, combinando visão computacional, sensoriamento 3D e aprendizado de máquina para tomada de decisão autônoma (Mueller et al., 2025; Lin et al., 2025). Com isso, a Indústria 4.0 não se limita à automação, mas transforma o chão de fábrica em um ambiente inteligente e resiliente, altamente responsivo às demandas operacionais e produtivas.

2.2 AGV

Os Automated Guided Vehicles (AGVs) são veículos autônomos projetados para realizar o transporte de materiais dentro de ambientes industriais, logísticos e hospitalares. Desde sua introdução na década de 1950, esses sistemas têm evoluído significativamente, passando de simples veículos guiados por trilhas magnéticas ou físicas para robôs autônomos capazes de navegar com base em mapas digitais, sensores e algoritmos de controle sofisticados (Ryck, 2020).

Os AGVs tradicionalmente operam em ambientes estruturados, seguindo rotas pré-definidas, com controle centralizado e limitado poder de adaptação. No entanto, a crescente demanda por flexibilidade e autonomia em sistemas logísticos levou ao desenvolvimento de AGVs mais inteligentes, capazes de interagir com o ambiente em tempo real e tomar decisões autônomas com base em dados sensoriais. Essa evolução foi impulsionada pela integração de tecnologias como sensores LiDAR, câmeras RGB-D, redes sem fio de alta velocidade, algoritmos de controle em tempo real e, mais recentemente, visão computacional baseada em aprendizado profundo (Lin et al., 2025; Golroudbari e Sabour, 2023).

Uma das principais vantagens dos AGVs está na automação segura e eficiente do transporte interno de cargas, reduzindo a dependência de operadores humanos e minimizando erros operacionais. Eles são utilizados em linhas de produção, almoxarifados e centros de distribuição, otimizando o fluxo de materiais, reduzindo custos operacionais e aumentando a rastreabilidade dos processos. Sua aplicação é particularmente valiosa em tarefas repetitivas e em ambientes hostis, onde há risco à integridade física de trabalhadores.

Figura 2.1: Exemplo de um AGV rebocador.

Contudo, apesar dos benefícios, AGVs também enfrentam desafios importantes. Um dos principais obstáculos está relacionado à rigidez de navegação em ambientes parcialmente estruturados ou com interferência humana. A falha na coleta de objetos, como pallets mal posicionados, ainda exige intervenções manuais frequentes, o que compromete o desempenho do sistema automatizado como um todo. Outro desafio é a integração eficiente entre percepção do ambiente, controle dinâmico e planejamento de trajetória em tempo real, exigindo o uso de sensores precisos e algoritmos robustos para garantir confiabilidade mesmo em condições adversas (Fraifer et al., 2024).

Recentemente, as pesquisas vêm focando no desenvolvimento de AGVs com capacidades aprimoradas de percepção e raciocínio autônomo, especialmente com o uso de visão computacional, redes neurais e controle adaptativo. Esses sistemas buscam romper com a limitação de rotas fixas, sendo capazes de identificar objetos, desviar de obstáculos, adaptar sua rota com base no estado do ambiente e interagir com operadores humanos de forma segura e fluida. Com isso, os AGVs evoluem de meros veículos guiados para verdadeiros robôs móveis autônomos, desempenhando papéis centrais em ambientes inteligentes (Nguyen et al., 2023; Mueller et al., 2025).

Além da percepção, outra área em expansão no contexto dos AGVs é a interação homem-máquina (HMI), fundamental para ambientes colaborativos. Sistemas de sinalização visual, comandos por gestos ou interfaces adaptativas permitem que operadores humanos instruam ou interrompam o AGV com facilidade, aumentando a segurança e a aceitação da tecnologia no chão

de fábrica. Também é crescente o uso de AGVs em configuração multi-robô, com coordenação centralizada ou distribuída para otimizar o fluxo logístico em grandes plantas industriais, o que demanda técnicas de alocação de tarefas, balanceamento de carga e planejamento cooperativo.

Por fim, com o avanço da computação embarcada e o uso de plataformas como ROS (Robot Operating System), tornou-se viável integrar visão computacional, sensores de profundidade e algoritmos de navegação em tempo real em AGVs de baixo custo. Essa democratização tecnológica tem ampliado o alcance dos AGVs em empresas de médio e pequeno porte, além de facilitar a experimentação e o desenvolvimento de novas funcionalidades por equipes técnicas internas.

2.3 AGV NA INDÚSTRIA 4.0

A transição da automação tradicional para os sistemas inteligentes da Indústria 4.0 impactou diretamente o papel dos AGVs nas fábricas e centros logísticos modernos. Enquanto anteriormente os AGVs funcionavam de forma isolada, operando sobre rotas fixas com lógica simples de navegação, o paradigma da Indústria 4.0 exige desses veículos um comportamento mais dinâmico, adaptativo e colaborativo (Schwab, 2017; Campilho e Silva, 2023).

Na Indústria 4.0, os AGVs são reconfigurados como agentes ativos de redes ciber-físicas, integrando-se a sistemas de planejamento de produção, controle de estoque e plataformas de análise de dados em tempo real. Essa integração é possibilitada por meio de redes industriais (ex: OPC UA, MQTT), sensores conectados, gateways IoT e software de gerenciamento centralizado. Assim, o AGV deixa de ser apenas um executor de tarefas logísticas e passa a colaborar com os demais elementos da cadeia produtiva, comunicando-se com sistemas *Enterprise Resource Planning* (ERP), *Manufacturing Execution System* (MES) e *Warehouse Management System* (WMS) para receber e relatar informações de sua operação em tempo real.

Além da conectividade, a Indústria 4.0 impõe demandas por maior flexibilidade operacional. Isso significa que o AGV precisa ser capaz de lidar com ambientes parcialmente estruturados, trajetórias variáveis, obstáculos imprevistos e tarefas que mudam dinamicamente. Nesse sentido, os sistemas de navegação baseados apenas em rotas estáticas tornaram-se insuficientes, abrindo espaço para tecnologias mais avançadas, como navegação *Simultaneous Localization and Mapping* (SLAM), localização por visão computacional, e fusão de sensores com base em aprendizado de máquina (Fraifer et al., 2024; Mueller et al., 2025).

O papel dos AGVs também se expande na perspectiva da eficiência e da adaptabilidade. Ao monitorar o estado do ambiente por meio de câmeras 3D, LiDARs ou sensores de corrente e temperatura, o AGV consegue identificar anomalias, tomar decisões localmente e evitar a paralisação da operação. Essa autonomia local, porém, deve ser balanceada com a coordenação global da fábrica, o que exige arquiteturas híbridas, combinando controle centralizado com inteligência distribuída embarcada nos veículos (Lin et al., 2025).

Outro ponto de destaque na integração dos AGVs à Indústria 4.0 é a rastreabilidade. Cada deslocamento, parada, coleta ou falha registrada pelo AGV pode ser sincronizada com sistemas em nuvem e utilizada para análise posterior. Isso viabiliza estratégias de manutenção preditiva, análise de produtividade e detecção de gargalos logísticos. Em muitos casos, os dados dos AGVs também alimentam gêmeos digitais (digital twins) da planta, permitindo simulações em tempo real para apoiar decisões operacionais e estratégicas.

É importante destacar ainda que o avanço de plataformas como o ROS Industrial e os pacotes compatíveis com os padrões da indústria possibilitam o uso de soluções modulares, escaláveis e interoperáveis. Isso facilita tanto o desenvolvimento de novos recursos quanto a integração com sistemas legados já existentes na planta.

Nesse cenário, o AGV torna-se não apenas um veículo de transporte, mas um elemento inteligente e adaptável dentro da fábrica conectada. Seu desempenho, sua capacidade de interação com operadores e sua habilidade de se adaptar a situações imprevistas, tornam-se diferenciais competitivos em um ambiente industrial cada vez mais automatizado e orientado por dados.

2.4 IA E VISÃO COMPUTACIONAL

A aplicação de Inteligência Artificial (IA) em sistemas robóticos tem revolucionado a forma como máquinas interagem com o ambiente ao seu redor. No contexto da Indústria 4.0, a IA permite que robôs móveis, como AGVs, realizem tarefas com um nível elevado de autonomia, adaptabilidade e robustez. Dentre as áreas mais relevantes para a autonomia robótica, destaca-se a visão computacional, que fornece aos robôs a capacidade de perceber visualmente o ambiente, identificar objetos, estimar distâncias e tomar decisões em tempo real com base nas informações extraídas das imagens (Terven et al., 2023; Lin et al., 2025).

A visão computacional baseada em IA, especialmente por meio de redes neurais convolucionais (CNNs), vem sendo amplamente adotada em tarefas industriais como inspeção de qualidade, monitoramento de segurança, controle de processos e, mais recentemente, na percepção e navegação de AGVs. Diferente de abordagens tradicionais baseadas em regras rígidas de processamento de imagem, os modelos baseados em aprendizado profundo são capazes de generalizar, adaptando-se a diferentes cenários, condições de iluminação e variações nos objetos a serem detectados (Mueller et al., 2025; Fraifer et al., 2024).

Modelos como o *You Only Look Once* (YOLO) exemplificam essa evolução ao permitir detecção de objetos em tempo real com alta precisão. O YOLOv8, em particular, introduz melhorias em arquitetura, segmentação de instâncias e eficiência computacional, tornando-o especialmente adequado para aplicações embarcadas em AGVs, onde a velocidade de inferência é tão crítica quanto a acurácia (Ultralytics, 2023; Sapkota et al., 2024). Além disso, o suporte nativo a segmentação de instâncias permite que não apenas a presença de objetos seja reconhecida, mas também suas formas exatas e contornos, o que é crucial para estimar a orientação e a posição de um pallet em uma cena tridimensional.

Combinada a sensores como câmeras RGB-D ou estéreo, a visão computacional possibilita a construção de mapas tridimensionais do ambiente, facilitando a navegação e a tomada de decisão baseada em percepção visual. A detecção de pallets, por exemplo, deixa de depender de marcadores físicos ou coordenadas fixas, e passa a ser baseada na própria forma e aparência do objeto detectado em tempo real. Isso torna o AGV mais robusto a falhas humanas, como o posicionamento incorreto do pallet fora da área previamente planejada.

O treinamento de modelos de visão computacional requer, no entanto, a disponibilidade de grandes quantidades de dados anotados, o que será abordado na próxima seção. Ainda assim, a reutilização de modelos pré-treinados em datasets genéricos como COCO ou ImageNet tem-se mostrado eficaz como ponto de partida, especialmente quando combinada a técnicas de fine-tuning com imagens capturadas no ambiente real de operação (Bochkovskiy et al., 2020; DATALABELIFY, 2023).

Em resumo, a integração entre IA e visão computacional oferece aos AGVs a capacidade de percepção ativa, permitindo que adaptem sua estratégia de coleta de pallets em tempo real, mesmo diante de falhas humanas ou variações do ambiente. Essa combinação representa um dos pilares tecnológicos mais importantes para garantir autonomia, flexibilidade e confiabilidade em ambientes industriais conectados.

2.5 BASE DE DADOS DE TREINAMENTO

A construção de bases de dados anotadas é uma etapa essencial no desenvolvimento de sistemas de visão computacional baseados em aprendizado profundo. Diferentemente dos algoritmos tradicionais, que funcionam com regras fixas de processamento de imagem, os modelos modernos de detecção e segmentação de objetos aprendem diretamente com os dados, extraindo padrões visuais a partir de milhares de exemplos rotulados. Por isso, a qualidade, a diversidade e a representatividade dos dados de treinamento são fatores determinantes para o desempenho do modelo em aplicações do mundo real (Bochkovskiy et al., 2020; DATALABELIFY, 2023).

Em contextos industriais, como o de AGVs operando em ambientes logísticos, datasets genéricos como COCO ou ImageNet não são suficientes para representar as variações específicas de iluminação, posicionamento, perspectiva e obstáculos encontrados em fábricas reais. Isso torna necessário o uso de datasets personalizados, construídos a partir de imagens capturadas no próprio ambiente de operação, contendo os objetos de interesse em diferentes posições, ângulos, distâncias e condições ambientais (Sapkota et al., 2024; Ultralytics, 2023).

A criação de datasets especializados envolve um processo de coleta e anotação manual de imagens. A ferramenta CVAT (Computer Vision Annotation Tool) tornou-se uma das plataformas mais utilizadas para essa finalidade, por ser gratuita, de código aberto e compatível com diversos formatos de saída, como YOLO, COCO, VOC e Mask R-CNN. O CVAT permite anotar bounding boxes, máscaras de segmentação e até keypoints, possibilitando a criação de conjuntos de dados adequados tanto para detecção de objetos quanto para segmentação de instâncias, como requerido pelo YOLOv8 (V. et al., 2020; TENSORGIRL, 2023).

Durante a anotação, é importante garantir diversidade nas imagens, variando posição dos pallets, distância da câmera, iluminação e presença de ruídos visuais, para que o modelo seja capaz de generalizar. Outro cuidado importante é balancear as classes de interesse, evitando o problema de *class imbalance*, que pode fazer com que o modelo apresente viés para determinadas categorias ou condições (Buda et al., 2018). Além disso, técnicas de *data augmentation*, como rotação, corte, variação de brilho e distorções geométricas, são amplamente aplicadas para aumentar a variabilidade dos exemplos sem a necessidade de coletar milhares de novas imagens.

Após a anotação, os dados devem ser organizados em conjuntos de treinamento, validação e teste, normalmente seguindo proporções como 70/15/15 ou 80/10/10. Essa separação é fundamental para garantir a imparcialidade na avaliação do modelo, permitindo que seu desempenho seja testado em dados que ele nunca viu durante o treinamento. Métricas como Precision, Recall e mAP (*mean Average Precision*) são comumente utilizadas para essa avaliação, especialmente em tarefas de detecção em tempo real (Terven et al., 2023).

Por fim, vale destacar que, embora existam iniciativas públicas de datasets industriais com pallets e empilhadeiras, muitas empresas preferem construir bases de dados privadas, pois isso garante a representação fiel do seu ambiente e dos desafios específicos enfrentados no chão de fábrica. Essa abordagem personalizada tem-se mostrado mais eficaz, especialmente quando combinada com modelos como o YOLOv8, que oferecem flexibilidade para treinamento e adaptação ao cenário real de aplicação.

2.6 CÂMERA 3D

Enquanto a visão computacional tradicional, baseada em imagens RGB, é eficaz para a detecção e classificação de objetos, ela apresenta limitações importantes em aplicações que requerem a compreensão da geometria e da profundidade do ambiente. Em cenários industriais, como a coleta autônoma de pallets por AGVs, é essencial que o sistema não apenas reconheça a

presença do objeto, mas também estime sua posição e orientação tridimensional em relação ao robô. Para esse objetivo, o uso de câmeras 3D tornou-se uma tecnologia fundamental (Hu et al., 2020; Fraifer et al., 2024).

As câmeras 3D (também conhecidas como sensores RGB-D) combinam imagem colorida com medição de profundidade em cada pixel. Elas são capazes de gerar uma nuvem de pontos do ambiente, ou seja, uma representação tridimensional que contém a localização espacial de milhares de pontos na cena. Modelos populares no contexto de robótica móvel incluem a Intel RealSense D435, a Orbbec Astra e a ZED Stereo Camera, amplamente adotadas devido à sua alta precisão, suporte a integração com ROS e compatibilidade com algoritmos de processamento de nuvem de pontos como Point Cloud Library (PCL) e Open3D.

Em tarefas como a coleta de pallets, as informações fornecidas pela câmera 3D permitem ao AGV estimar com precisão a distância até o objeto, o desvio lateral em relação à sua linha central, a inclinação do pallet, e até mesmo verificar se o objeto está em uma posição fisicamente acessível. Essa percepção de profundidade é essencial para alinhar o robô de forma eficaz antes da coleta, especialmente quando o posicionamento do pallet foi feito de maneira incorreta por operadores humanos, uma situação frequente em ambientes reais.

Além disso, a nuvem de pontos pode ser utilizada para gerar mapas locais do entorno, identificar obstáculos em 3D e calcular planos de aproximação seguros. Em cenários mais avançados, pode-se empregar algoritmos de pose estimation para ajustar em tempo real a posição e o ângulo de ataque do robô com base na forma do objeto detectado. Quando integrada a um modelo de detecção como o YOLOv8, a profundidade adiciona uma dimensão extra à percepção: o reconhecimento da instância do objeto é enriquecido com sua posição no espaço.

Outro benefício da câmera 3D é sua capacidade de operar mesmo em condições de iluminação desafiadoras. Enquanto câmeras RGB tradicionais dependem fortemente da luz ambiente, sensores de profundidade baseados em infravermelho ou estéreo podem funcionar com mais robustez em ambientes industriais mal iluminados ou com reflexos metálicos, típicos de fábricas e depósitos logísticos.

A integração de sensores RGB-D com sistemas embarcados em AGVs é favorecida pelo suporte nativo em middlewares como o Robot Operating System (ROS), que oferece drivers, mensagens e ferramentas de visualização compatíveis. Essa compatibilidade facilita o desenvolvimento de pipelines completos de percepção e controle baseados em profundidade, desde a aquisição de dados até a tomada de decisão autônoma do robô.

Por fim, com a redução dos custos e a miniaturização dos sensores 3D, tornou-se viável embarcar essas tecnologias mesmo em AGVs compactos ou de menor custo. Essa acessibilidade tem permitido que soluções industriais cada vez mais completas e inteligentes sejam desenvolvidas internamente por equipes técnicas, sem a necessidade de grandes investimentos em hardware de alto desempenho.

2.7 CONTROLE DE AGV

Para que um AGV execute com sucesso tarefas como a coleta precisa de pallets, não basta apenas detectar e localizar o objeto, é necessário que o sistema de controle seja capaz de atuar dinamicamente sobre os motores de tração e direção, realizando ajustes finos de posição e orientação em tempo real. Esse processo é conduzido por sistemas de controle embarcados que, com base nas informações de sensores e visão computacional, calculam os comandos de movimento apropriados para guiar o robô até sua meta de forma segura e eficiente (Wang et al., 2017; Åström e Murray, 2008).

Dentre os métodos de controle mais utilizados na robótica móvel, destaca-se o controlador *Proportional-Integral-Derivative* (PID). Trata-se de um algoritmo clássico, amplamente adotado por sua simplicidade e eficácia, que calcula o erro entre uma variável de referência (como o centro de um pallet detectado) e o valor atual medido (posição do AGV), e aplica correções com base em três termos: proporcional ao erro atual, acumulado ao longo do tempo (integral), e sua taxa de variação (derivativo).

Em aplicações industriais com AGVs, o PID é comumente utilizado para alinhamento lateral ao pallet, controle de velocidade ao se aproximar da carga, e correção angular para garantir que a garra ou garfos estejam orientados corretamente antes da coleta. Esses ajustes finos são particularmente importantes em ambientes onde o posicionamento dos pallets pode variar devido a falhas humanas ou falta de marcações fixas, como ocorre frequentemente na prática.

A eficácia de um controlador PID depende fortemente de sua sintonia. Ganhos inadequados podem levar a oscilações, lentidão ou instabilidade, especialmente quando o AGV opera em tempo real com informações vindas de sensores com ruído ou atraso. Por isso, diversas estratégias de ajuste foram desenvolvidas ao longo do tempo, incluindo métodos empíricos como *Ziegler-Nichols* e técnicas mais modernas baseadas em otimização automática (Kim et al., 2019; Abajo et al., 2022).

Em sistemas integrados com visão computacional, como no caso deste trabalho, o controlador PID pode utilizar diretamente o erro visual calculado a partir da imagem: por exemplo, a diferença entre o centro do pallet detectado pelo YOLOv8 e o centro da imagem da câmera. Essa abordagem, conhecida como *visual servoing*, permite que o robô reaja dinamicamente à posição visual do objeto, sem depender exclusivamente de coordenadas absolutas.

Além disso, o controlador pode atuar em conjunto com dados de profundidade para ajustar a aproximação do robô, garantindo que a coleta ocorra apenas quando o pallet estiver a uma distância adequada e sem obstáculos à frente. A fusão dessas informações visuais e espaciais com o controle é fundamental para garantir precisão e segurança, especialmente em ambientes industriais com tráfego compartilhado e movimentação imprevisível.

O PID também pode ser estendido para arquiteturas mais complexas de controle, como controle hierárquico, em que a camada superior define o comportamento geral (por exemplo, iniciar aproximação ao pallet) e a camada inferior realiza os ajustes finos (por exemplo, corrigir deslocamento lateral). Essa separação permite uma operação mais modular e facilita a integração com sistemas de alto nível, como o planejador de tarefas do AGV ou um orquestrador central de logística.

Por fim, o uso de controladores clássicos como o PID continua extremamente relevante mesmo diante de algoritmos mais recentes baseados em aprendizado de máquina, principalmente devido à sua confiabilidade, interpretabilidade e baixo custo computacional, características que são ideais para sistemas embarcados em tempo real.

2.8 CONCLUSÃO

Este capítulo apresentou os conceitos que sustentam o trabalho: (i) Indústria 4.0 e o papel dos AGVs em cenários conectados, (ii) capacidades e limitações dos AGVs, (iii) fundamentos de IA/visão computacional com ênfase em segmentação por instâncias, (iv) construção de bases de dados industriais e critérios de particionamento, (v) sensores RGB-D e (vi) controle de baixo nível via PID. Esses elementos embasam as decisões de projeto adotadas na metodologia do Cap. 3, em especial o uso de YOLOv8-seg para extração de máscaras, a combinação com profundidade para pose do pallet e a integração com `tf2_ros` e controle PID para navegação corrigida.

3 METODOLOGIA

A metodologia deste trabalho foi estruturada para resolver um problema real observado no ambiente industrial: a dificuldade de coleta autônoma de pallets mal posicionados por AGVs. Para isso, adotou-se um pipeline completo que integra visão computacional, processamento de profundidade, transformações espaciais no ROS e geração dinâmica de pontos de navegação.

As etapas foram organizadas de forma sequencial e iterativa, iniciando pela coleta de dados no ambiente real, seguida da criação de um dataset segmentado, treinamento do modelo YOLOv8-seg, cálculo da profundidade dos pontos-chave do pallet e projeção da pose no mapa global do AGV. Por fim, essas informações foram utilizadas para gerar trajetórias corrigidas de navegação, permitindo a coleta mesmo quando o pallet está desalinhado.

Este capítulo descreve detalhadamente cada etapa do processo, bem como as decisões técnicas que permitiram integrar as componentes de visão, localização e controle no sistema final.

3.1 VISÃO GERAL DA METODOLOGIA

O desenvolvimento deste trabalho seguiu uma abordagem prática e iterativa, baseada na observação de um problema real em ambiente industrial e na implementação de uma solução baseada em visão computacional e controle robótico. O objetivo principal foi permitir que um AGV fosse capaz de detectar e coletar pallets de forma autônoma, mesmo quando estes estivessem mal posicionados por operadores humanos, algo frequente no cotidiano das empresas.

A metodologia foi dividida em sete etapas principais:

1. Aquisição de imagens com a câmera MRDVS posicionada na frente do AGV, mirando perpendicularmente ao chão;
2. Extração de frames e anotação das máscaras de pallets no CVAT;
3. Treinamento do modelo YOLOv8-seg com segmentação de instâncias;
4. Extração de três pontos do pallet (esquerda, centro, direita) e cálculo da profundidade;
5. Projeção da pose do pallet no referencial global usando as transformações do ROS;
6. Geração de pontos de navegação sobre o eixo do pallet corrigido;
7. Execução da manobra de alinhamento e coleta com base nesses pontos.

3.2 AQUISIÇÃO DE IMAGENS COM A CÂMERA MRDVS

A primeira etapa prática do projeto consistiu na aquisição de imagens reais de pallets em ambiente industrial, utilizando a câmera Lanxin MRDVS S2, um equipamento que combina captura de imagem RGB com dados de profundidade (*Depth Map*) em tempo real. Esta câmera foi escolhida por sua robustez e integração com sistemas embarcados, sendo amplamente utilizada em aplicações de navegação autônoma e visão computacional industrial.

Figura 3.1: Câmera Lanxin MRDVS S2 Max utilizada no AGV, equipada com sensores RGB e profundidade para navegação e percepção frontal.

A visualização e as especificações técnicas da câmera MRDVS S2 Max utilizadas neste trabalho podem ser observadas na Figura 3.1 e na Figura 3.2.

Modelo	Câmera de prevenção de obstáculos S2 Max
Iluminação	940nm VCSEL
saída	Mapas de amplitude de profundidade/RGB/IR
Resolução de profundidade (H × V)	320 × 240 px
Taxa de quadros de profundidade	Máx. 15 fps, normalmente 12 fps
Campo de visão de profundidade (H × V)	81 ° × 61 °
Resolução RGB (H × V)	1920 × 1080 px
Taxa de quadros RGB	Máx. 15 fps, típico 12 fps
Campo de visão RGB (H × V)	88 ° × 56 °
Tipo de obturador RGB	Shutter
Área de trabalho	0.2 m ~ 5 m
Precisão	±3 mm + 0.5% × profundidade
Consumo de energia	6.4 W
Dimensões (C × L × A)	92 mm × 47 mm × 51 mm
Peso	449 g
Fonte de alimentação do laboratório	24 V DC / 2 A
Comunicação de dados	Gigabit Ethernet e 4 canais de E/S*
Classificação do gabinete	IP67
Temperatura de Operação	-20 ° C ~ 60 ° C
Temperatura de armazenamento	-25 ° C ~ 85 ° C
Linguagem de Programação	SDK C/C++/ROS1/ROS2
Sistema Operacional	ARM Linux (AArch64), Linux (x86_64), Windows 7/8/10/11

Figura 3.2: Especificações técnicas da câmera Lanxin MRDVS S2 Max, incluindo faixa de profundidade, resoluções disponíveis e parâmetros de operação.

Para melhorar a precisão da detecção e da estimativa de profundidade, a câmera foi instalada na parte frontal do AGV, posicionada no centro do garfo da empilhadeira, com o sensor voltado para frente e orientado perpendicularmente ao chão (ângulo de 90°). Essa posição estratégica permite capturar diretamente os pallets no campo de visão frontal, simulando a perspectiva real de coleta durante a aproximação do robô.

A posição exata da câmera MRDVS no AGV, instalada acima dos garfos e voltada perpendicularmente ao solo, é mostrada na Figura 3.3.

Figura 3.3: Posicionamento da câmera MRDVS no AGV: instalada centralmente acima dos garfos, orientada perpendicularmente ao solo para maximizar a precisão da segmentação e da profundidade. (Foto produzida pelo autor.)

As resoluções escolhidas para as imagens RGB (960×540) e de profundidade (320×240, com binning 2×2 ativado) foram definidas com base em testes práticos, visando atingir o

melhor equilíbrio possível entre qualidade de segmentação e tempo de execução. Resoluções maiores apresentaram ganho marginal na acurácia visual, mas comprometeram o desempenho em tempo real, especialmente no ambiente embarcado. Já resoluções muito baixas impactavam negativamente a definição das máscaras e a precisão na estimativa de profundidade. O formato adotado se mostrou adequado tanto para o treinamento do modelo quanto para sua execução contínua a bordo do AGV.

Com a câmera montada, foram realizadas diversas gravações em ambientes reais da empresa, onde o AGV opera diariamente. Os vídeos foram gravados em diferentes condições, contemplando:

- Variações de cor dos pallets (plástico azul, preto e branco/preto.);
- Diferentes ângulos de posicionamento dos pallets em relação ao robô (alinhado, inclinado, descentralizado);
- Condições de iluminação variadas, como luz natural, artificial e sombra;
- Presença de ruídos visuais, como obstáculos no fundo, chão reflexivo e variações de textura.

Inicialmente, foram realizadas gravações com pallets reais utilizados pelo cliente final, os quais foram enviados para a empresa com o objetivo de realizar testes internos. Essas coletas serviram para validar a abordagem proposta neste trabalho, bem como para testar outras funcionalidades relacionadas ao sistema do AGV. Após a validação e ajustes iniciais, uma nova etapa de aquisição foi realizada diretamente no ambiente da fábrica do cliente, já com o sistema operando próximo das condições reais de uso. As imagens obtidas nessa segunda fase foram as únicas utilizadas para a construção do dataset final que alimentou o treinamento do modelo YOLOv8-seg em produção.

O objetivo dessa etapa foi construir uma base de imagens realista e diversificada, representando as situações mais comuns, e também os desafios, enfrentados pelo AGV em seu ambiente de trabalho. Esses vídeos formaram o material bruto que seria posteriormente utilizado na geração do dataset e no treinamento do modelo de detecção.

A próxima etapa foi a extração de frames desses vídeos e a segmentação das máscaras dos pallets, conforme descrito na seção seguinte.

3.3 ANOTAÇÃO NO CVAT E CRIAÇÃO DO DATASET

Com os vídeos capturados pela câmera MRDVS em diferentes situações, a etapa seguinte consistiu na extração de frames relevantes que representassem bem a diversidade de casos encontrados no ambiente industrial. Esses frames foram salvos como imagens estáticas e organizados para compor o dataset personalizado necessário para o treinamento do modelo de segmentação.

Para a anotação das imagens, foi utilizada a ferramenta CVAT (Computer Vision Annotation Tool) (V. et al., 2020), amplamente adotada em projetos de visão computacional por sua flexibilidade, suporte a múltiplos formatos e interface amigável. O tipo de anotação escolhido foi a segmentação de instâncias, em que cada pallet visível na imagem é contornado manualmente, pixel a pixel, para gerar uma máscara individual.

A Figura 3.4 apresenta a interface da ferramenta CVAT utilizada no processo de anotação das máscaras de segmentação.

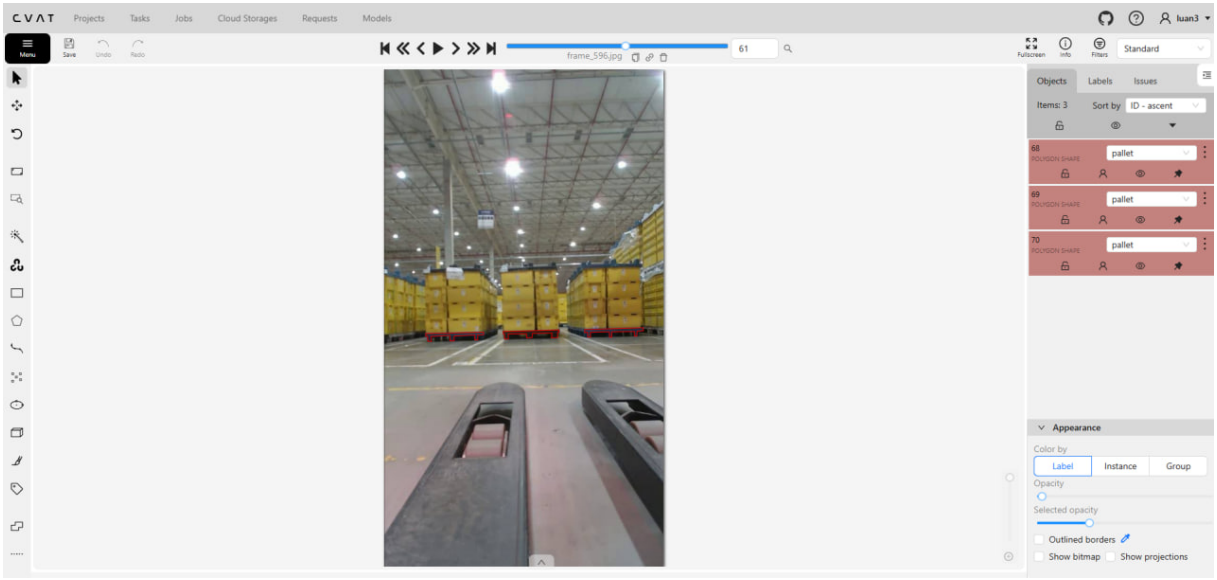


Figura 3.4: Interface da ferramenta CVAT utilizada para anotação manual das máscaras de pallets no dataset.

Essa abordagem é mais precisa do que caixas delimitadoras (bounding boxes), pois permite capturar a forma real do objeto, algo essencial para:

- Calcular o centro real da massa do pallet;
- Determinar as extremidades esquerda e direita da máscara para medição de profundidade;
- Estimar a orientação angular do pallet no campo de visão.

O processo de anotação manual levou em consideração:

- Casos com múltiplos pallets na mesma cena;
- Diferenças angulação, cor e iluminação;
- Ruídos no fundo (pontos reflexivos, manchas).

Após a conclusão da segmentação, as anotações foram exportadas no formato YOLOv8-seg, compatível com a biblioteca Ultralytics utilizada no treinamento. As máscaras foram salvas como arquivos .txt contendo as coordenadas poligonais normalizadas e associadas às imagens .jpg correspondentes.

O dataset completo foi então dividido em três subconjuntos:

- Conjunto de treinamento (train): 80% das imagens, utilizadas para ajustar os pesos da rede neural.
- Conjunto de validação (val): 10% das imagens, usado para monitorar o desempenho do modelo durante o treinamento.
- Conjunto de teste (test): 10% das imagens, reservado para avaliar o modelo de forma imparcial após o treinamento.

Essa divisão foi feita com base na literatura TensorGirl (2023) (TENSORGIRL, 2023), garantindo que as variações de cenário estivessem representadas em todos os conjuntos. A qualidade da anotação e a diversidade das imagens são fatores decisivos para o sucesso do treinamento do modelo YOLOv8, tema que será abordado na próxima seção.

3.4 TREINAMENTO DO MODELO YOLOV8-SEG

Para realizar a segmentação precisa de pallets em imagens reais capturadas pelo AGV, foi utilizado o modelo YOLOv8n-seg (versão nano com suporte à segmentação de instâncias), disponibilizado pela Ultralytics. A escolha dessa variante se deu pela necessidade de obter baixa latência e alta compatibilidade com sistemas embarcados, sem comprometer significativamente a precisão do modelo.

O processo de treinamento foi dividido em duas etapas. Inicialmente, foi treinado um modelo com um conjunto de aproximadamente 150 imagens capturadas internamente na empresa, utilizando pallets reais enviados pelo cliente. Essa primeira versão teve como objetivo validar a viabilidade da abordagem e ajustar os parâmetros iniciais de detecção.

Após os testes internos, foi realizada uma nova coleta de imagens diretamente no ambiente do cliente final, totalizando aproximadamente 335 imagens segmentadas manualmente no CVAT. Esse segundo conjunto de dados, mais próximo da realidade de operação, foi utilizado para treinar o modelo final que foi embarcado no sistema de visão do AGV. Novas imagens seguem sendo adicionadas ao dataset à medida que novos pallets e condições operacionais são encontrados, permitindo a revalidação contínua do modelo.

Para evitar *data leakage*, os frames foram particionados por vídeo (e não por frame), garantindo que cenas adjacentes não aparecessem em subconjuntos distintos.

3.4.1 Configurações do treinamento

O modelo foi treinado utilizando a interface de linha de comando (CLI) da biblioteca Ultralytics, com os seguintes parâmetros principais:

- Modelo: yolov8n-seg
- Tamanho das imagens (imgsz): padrão da Ultralytics (640×640)
- Épocas máximas: 1000
- Parâmetro de paciência (patience): 100 (early stopping ativado)
- Batch size: -1 (automático baseado na GPU)
- Aumento de dados (augmentation): padrão Ultralytics (flip, blur, brightness, crop, etc.)

O treinamento foi interrompido automaticamente pela estratégia de early stopping após 576 épocas, quando as métricas de validação estabilizaram.

Tanto no conjunto interno quanto no conjunto final, a divisão dos dados seguiu a proporção clássica de: 80% para treinamento, 10% para validação e 10% para teste. As imagens utilizadas estavam todas redimensionadas para 960×540 pixels, visando manter a proporção real da câmera MRDVS e reduzir a carga computacional durante a inferência.

Embora versões mais recentes da família YOLO (como YOLOv9, YOLOv10 e, mais recentemente, YOLOv11) tenham sido lançadas durante o desenvolvimento deste trabalho, optou-se pelo YOLOv8 por razões técnicas e operacionais.

Primeiramente, o desenvolvimento inicial da solução começou quando o YOLOv8 representava o estado da arte em desempenho e estabilidade, sendo amplamente adotado pela comunidade e totalmente compatível com o ecossistema de ferramentas da empresa. Assim, a equipe já possuía experiência prévia com essa arquitetura, o que reduziu significativamente o tempo de implementação, depuração e integração com o sistema embarcado do AGV.

Além disso, a empresa priorizou uma solução robusta e comprovada, evitando investir tempo em testes exploratórios com múltiplas versões mais recentes da YOLO que poderiam demandar ajustes adicionais nos pipelines de treinamento, conversão de modelos, otimizações para execução embarcada e compatibilidade com ROS.

Em suma, a escolha do YOLOv8 refletiu um compromisso entre maturidade tecnológica, baixo risco de integração e eficiência no desenvolvimento, garantindo que o projeto pudesse avançar rapidamente sem prejuízo de desempenho.

3.5 EXTRAÇÃO DA PROFUNDIDADE E PONTOS RELEVANTES DA MÁSCARA

Com o modelo YOLOv8-seg treinado e em funcionamento, o sistema passou a detectar, em tempo real, a segmentação precisa do pallet na imagem RGB. O objetivo dessa etapa, no entanto, não se restringe apenas à detecção visual, é fundamental obter informações geométricas que permitam reconstruir a orientação do pallet no ambiente real, especialmente nos casos em que ele estiver desalinhado ou fora da posição padrão esperada pelo sistema de navegação.

Para isso, a partir da máscara segmentada do pallet, o algoritmo extrai automaticamente três pontos relevantes:

- Extremidade esquerda da máscara (ponto mais à esquerda da projeção);
- Extremidade direita da máscara;
- Centro da máscara, estimado como a média entre os dois extremos.

Esses pontos são extraídos no domínio da imagem RGB e imediatamente convertidos para valores de profundidade com base na imagem de profundidade fornecida pela câmera MRDVS. Cada ponto, portanto, passa a ter não apenas sua posição 2D (em pixels), mas também sua distância real em metros até a câmera, o que permite projetá-lo no espaço tridimensional.

Além da extração dos pontos relevantes da máscara, foi necessário realizar um ajuste adicional devido à diferença de resolução entre os canais da câmera MRDVS. A imagem RGB utilizada na inferência possui resolução de 960×540 pixels, enquanto a imagem de profundidade, operando com binning 2×2, é reduzida para 320×240 pixels. Para que os pontos detectados na segmentação fossem corretamente mapeados para o canal de profundidade, foi aplicado um cálculo de escala proporcional nas coordenadas, considerando as razões de 1/3 no eixo horizontal e 4/9 no eixo vertical. Esse procedimento garantiu que os valores de profundidade extraídos correspondessem com precisão à posição dos pontos detectados no domínio visual, mantendo a consistência entre os canais da câmera.

Em algumas situações, observou-se que os pixels exatamente na borda do pallet, definidos pela máscara segmentada na imagem RGB, não possuíam valores válidos na imagem de profundidade correspondente. Esse efeito é comum em sensores RGB-D, especialmente em superfícies com variações de textura, reflexos ou ângulos oblíquos em relação à câmera, onde os limites visuais nem sempre coincidem com o mapeamento de profundidade. Para contornar essa limitação, foi implementada uma estratégia de busca local ao redor do ponto alvo, com uma janela de tolerância definida, percorrendo os vizinhos mais próximos até encontrar um pixel com valor de profundidade válido. Esse procedimento aumentou significativamente a robustez da extração de profundidade, principalmente nas extremidades esquerda e direita do pallet, onde falhas de leitura são mais frequentes.

Esses três pontos, combinando visão e profundidade, formam a base para o cálculo da pose do pallet, ou seja, sua posição (x, y) e sua orientação. Essa informação é essencial para

adaptar a trajetória do robô nos casos em que o pallet estiver fora do local esperado, como será descrito na próxima seção.

3.6 CÁLCULO DA POSE CORRIGIDA DO PALLET NO MAPA

A partir da detecção do pallet e da obtenção das distâncias e ângulos dos três pontos de referência (esquerda, centro e direita), inicia-se o processo de cálculo da pose corrigida do pallet. Essa etapa é responsável por transformar as informações métricas obtidas pela câmera em uma posição e orientação espacial completas no sistema global de coordenadas do AGV

Para isso, o nó ROS responsável pelo cálculo da pose realiza continuamente a escuta e publicação de transformações entre diferentes quadros de referência (frames) do robô, por meio das bibliotecas tf e tf2_ros. Essas transformações descrevem as relações espaciais entre componentes físicos do sistema, como a câmera, o corpo do AGV, o sensor LiDAR, e os sistemas de referência internos e externos, e são atualizadas em tempo real por diversas threads paralelas.

Dentre essas transformações, destacam-se:

- `base_link` → `camera_link` , que define o deslocamento físico e angular entre o centro geométrico do AGV e a câmera MRDVS;
- `mundo` → `reflexivo` → `map_scan` → `map` → `base_link` , que compõem a cadeia de transformações utilizada para localizar o robô no ambiente global com base nos espelhos refletores;
- `camera_link` → `pallet` , que define a posição e orientação do pallet em relação à câmera, com base nos pontos detectados e nas distâncias medidas.

O cálculo da pose começa com a conversão das distâncias e ângulos dos pontos esquerdo e direito, recebidos via tópicos ROS, em coordenadas cartesianas relativas à câmera

$$x = d\cos(\theta), \qquad y = d\sin(\theta).$$

. Em seguida, o ponto central é usado para estimar a posição média do pallet, sendo aplicado um filtro de média temporal com janela de dois segundos. Esse filtro, implementado no código como um somatório contínuo dentro da thread que faz o apontamento da câmera para o pallet, suaviza ruídos e pequenas flutuações das leituras da câmera, garantindo estabilidade nas coordenadas.

Com os três pontos conhecidos, é calculada a orientação do pallet (ângulo de yaw) a partir da reta que conecta os pontos esquerdo e direito, usando a relação:

$$\theta = \arctan\left(\frac{y_{dir} - y_{esq}}{x_{dir} - x_{esq}}\right) \tag{3.1}$$

Esse ângulo é então corrigido por um cálculo de vetor normal (theta_n), de modo que o pallet sempre esteja orientado para o AGV, evitando ambiguidade de direção. O resultado é uma orientação corrigida, que é usada para gerar transformadas tf adicionais para os frames pallet_esquerda, pallet_direita e pallet.

Por fim, o sistema utiliza o buffer de transformações (tf_buffer) para converter a posição do pallet detectado do frame da câmera (camera_link) para o frame global (mundo). A função lookup_transform busca a transformação válida e, caso falhe (por exemplo, devido a perda de sincronização ou ruído nos dados), o sistema aborta a operação para evitar comandos incorretos ao AGV. Quando a transformação é bem-sucedida, o ponto é convertido usando

tf2_geometry_msgs.do_transform_point, resultando nas coordenadas reais (x, y) do centro do pallet no mapa global. Esses valores são então publicados no tópico /pallet_xy e também visualizados no RViz através de markers esféricos, facilitando a validação e depuração do sistema durante os testes.

A relação entre os principais frames utilizados no cálculo da pose pode ser visualizada na Figura 3.5.

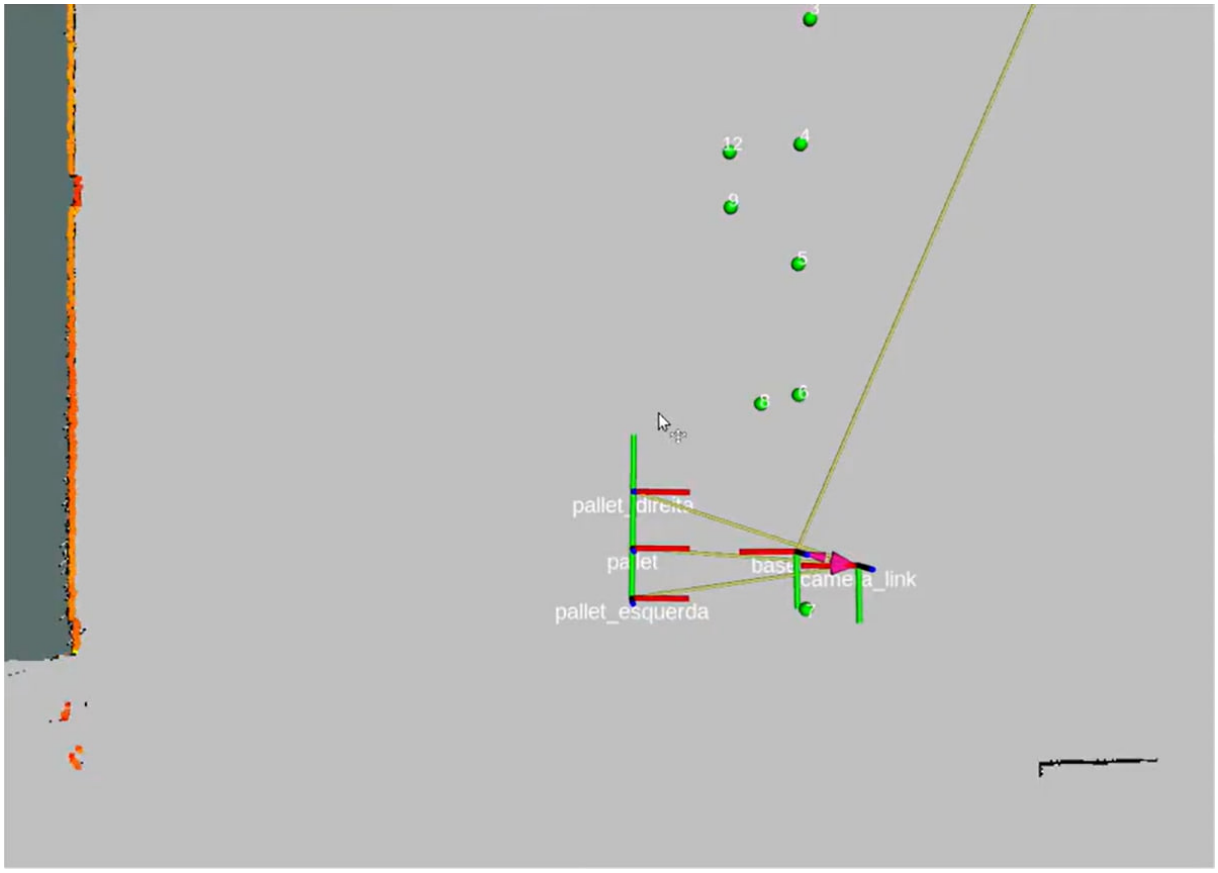


Figura 3.5: Visualização no RViz das transformações e frames relevantes (camera_link, pallet e mapa) durante a detecção e estimativa da pose.(Foto produzida pelo autor.)

Dessa forma, o resultado dessa etapa é a pose corrigida do pallet no mapa, composta por três elementos:

- x, y: posição no mapa global em metros;
- θ: orientação do pallet em relação ao AGV.

Essas informações constituem a base para a próxima etapa do sistema — a geração dinâmica dos pontos corrigidos de navegação.

3.7 GERAÇÃO DOS PONTOS CORRIGIDOS DE NAVEGAÇÃO

Com a pose do pallet corretamente estimada e transformada para o sistema de coordenadas global (frame mundo), o próximo passo do sistema consiste na geração de uma nova sequência de pontos de navegação que o AGV deverá seguir. Esses pontos corrigidos garantem que o robô

consiga se alinhar de forma precisa ao pallet, mesmo que ele esteja posicionado fora do ângulo ideal.

Esse processo é iniciado após o AGV realizar a aproximação lateral ao pallet e efetuar um giro de 90 graus, posicionando sua face frontal (com a câmera) diretamente voltada para o alvo. A câmera, já previamente calibrada e com transformações ativas, detecta o pallet, calcula sua posição central e orientação e, em seguida, inicia a lógica de criação da nova trajetória.

A geração dos pontos considera tanto a posição quanto a angulação estimada do pallet. Para isso, são utilizados parâmetros previamente definidos e armazenados em uma base de dados interna, que descrevem os deslocamentos em linha reta e em ângulo que o robô deve executar para se aproximar do pallet corretamente. Cada ponto da trajetória é definido com base em uma distância linear e um desvio lateral (em coordenadas relativas), e pode ainda ter funções associadas, como reduzir a velocidade, alinhar os garfos ou executar o empilhamento.

A Figura 3.6 apresenta os pontos corrigidos de navegação gerados automaticamente a partir da pose estimada do pallet. Os pontos verdes representam a trajetória que o AGV deve seguir até o alinhamento final, enquanto os pontos vermelhos indicam a trajetória corrigida em relação ao pallet, incluindo suas extremidades esquerda e direita e o centro calculado. Esses pontos servem como base para orientar o AGV durante a aproximação e garantir que ele se alinhe corretamente ao eixo do pallet

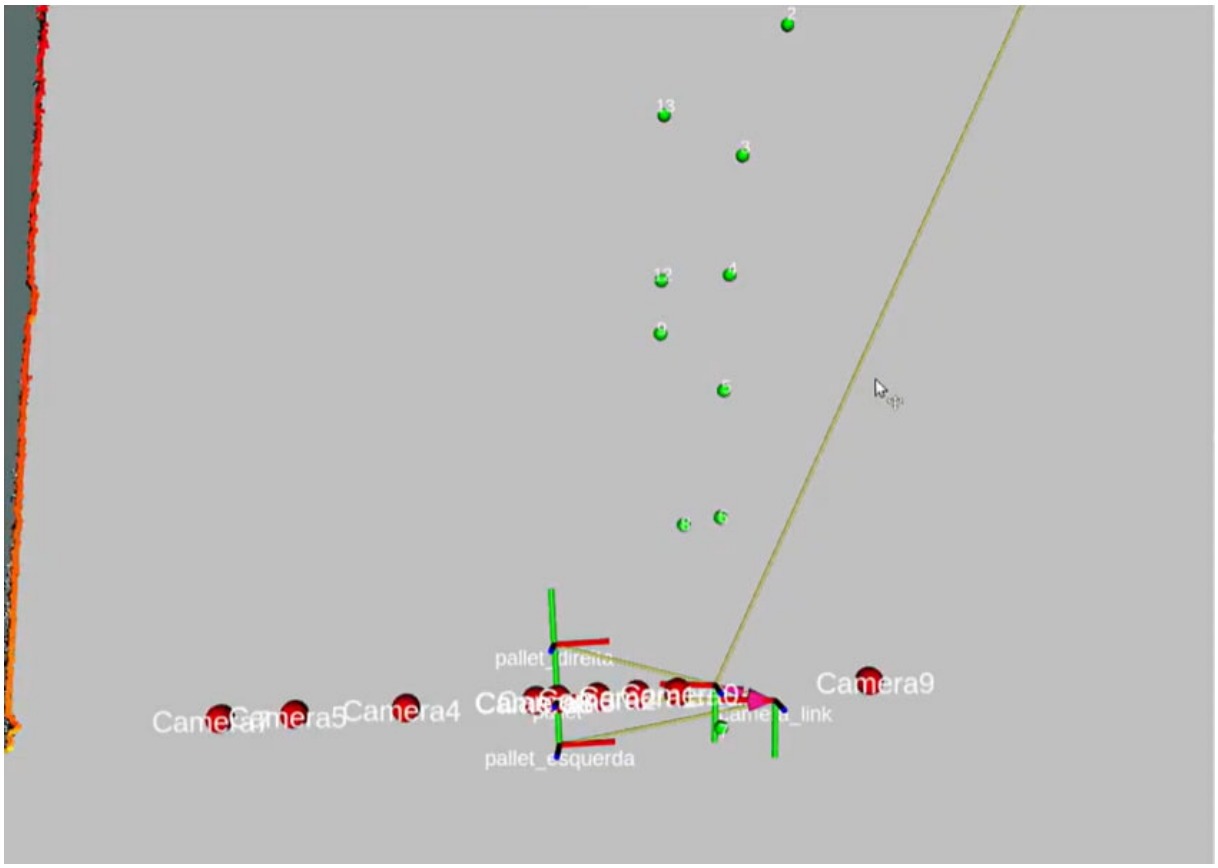


Figura 3.6: Visualização dos pontos corrigidos de navegação gerados no RViz com base na pose estimada do pallet. (Foto produzida pelo autor.)

A pose do pallet é usada como base para calcular essas posições. Para garantir estabilidade e robustez nas transformações espaciais, o sistema realiza múltiplas medições da pose ao longo de um intervalo de tempo (em torno de 3 segundos), e computa a média das posições

e orientações coletadas. Essa média evita que ruídos ou oscilações momentâneas prejudiquem o resultado, e garante que os pontos gerados estejam realmente alinhados à orientação do pallet.

Uma vez definida a sequência, os pontos são convertidos para o sistema de coordenadas global do mapa (mundo) por meio de transformações espaciais (usando a cadeia de transformações que parte da câmera até o mundo). Isso permite que o AGV, já localizado no mapa por meio de sistemas de localização baseados em espelhos refletores, consiga interpretar esses novos pontos como parte de sua rota e os siga naturalmente.

Se necessário, o sistema pode ainda gerar um ponto de retorno, levando em conta a posição anterior do AGV antes de iniciar a manobra. Esse ponto adicional permite que o robô volte ao ponto de origem após completar a tarefa, como, por exemplo, após empilhar o pallet e precisar retornar à sua trajetória original.

Durante a execução, o AGV segue os pontos corrigidos de forma contínua, acionando as funções programadas associadas a cada ponto. Com isso, é capaz de ajustar dinamicamente sua posição e orientação para realizar a aproximação final com precisão, garantindo que os garfos da empilhadeira encaixem corretamente sob o pallet. Caso os pallets estejam mal posicionados, o que é muito comum em ambientes industriais, o sistema é capaz de corrigir a trajetória automaticamente, eliminando a necessidade de intervenção manual por parte dos operadores.

Essa abordagem permite reaproveitar toda a estrutura de navegação já consolidada do AGV, mas adiciona uma camada de inteligência dinâmica que adapta a trajetória em tempo real. Na prática, o robô passa a ser capaz de realizar manobras de coleta e empilhamento com maior autonomia e confiabilidade, mesmo em situações onde os pallets se encontram desalinhados ou em posições imprecisas. Essa solução reduz significativamente o número de falhas operacionais e a dependência de correções manuais por parte da equipe de chão de fábrica, aumentando a eficiência do processo logístico como um todo.

3.8 FLUXOGRAMA

Figura 3.7: Fluxograma da metodologia. (Produzido pelo autor.)

3.9 CONSIDERAÇÕES FINAIS DA METODOLOGIA

A metodologia apresentada ao longo deste capítulo demonstrou como é possível integrar técnicas modernas de visão computacional, transformação espacial em sistemas robóticos e controle baseado em mapas para resolver um problema concreto do setor industrial: a coleta autônoma de pallets mal posicionados por um AGV.

A solução desenvolvida foi concebida para operar de maneira robusta em condições reais de fábrica, enfrentando desafios como variação de iluminação, desalinhamentos frequentes dos pallets, reflexos no chão e diferentes modelos e materiais de pallets. Por isso, cada etapa da metodologia foi cuidadosamente planejada e testada, desde a aquisição de imagens com a câmera MRDVS até a geração dos pontos corrigidos de navegação.

O processo de construção do dataset e treinamento do modelo YOLOv8-seg permitiu que o sistema identificasse com alta precisão os contornos dos pallets, mesmo em condições adversas. A extração dos três pontos estratégicos (esquerda, centro e direita) da máscara segmentada, associada à imagem de profundidade, viabilizou o cálculo da orientação real do pallet, uma informação crucial para adaptar a trajetória do robô.

Ao integrar essas informações com o sistema de transformações espaciais do ROS, o sistema foi capaz de converter a posição local do pallet (relativa à câmera) para uma pose global no mapa da fábrica, permitindo ao AGV reposicionar sua trajetória em tempo real. A lógica de geração de pontos corrigidos, baseada em parâmetros configuráveis no banco de dados, mostrou-se eficaz para alinhar o robô com o pallet e permitir a execução precisa da manobra de empilhamento.

Além disso, a adoção de filtros temporais e médias espaciais nas transformações contribuiu para a estabilidade do sistema, evitando que variações momentâneas nas leituras da câmera comprometessem a qualidade das decisões de navegação.

Por fim, ao manter a compatibilidade com a arquitetura de navegação já existente no AGV, a metodologia proposta adiciona um componente inteligente e adaptativo ao sistema, sem a necessidade de alterar profundamente os módulos de controle ou os planos de rota originais.

Como resultado, o AGV passou a ser capaz de operar com maior grau de autonomia, reduzindo intervenções manuais e falhas na coleta, e oferecendo uma solução eficiente, replicável e escalável para aplicações industriais que exigem manobras precisas em ambientes não estruturados.

4 RESULTADOS E DISCUSSÃO

4.1 VISÃO GERAL DOS EXPERIMENTOS

Os experimentos realizados neste trabalho tiveram como objetivo validar a eficácia do sistema proposto de detecção e posicionamento de pallets, bem como a integração entre o modelo de visão computacional, a câmera MRDVS e os módulos de navegação do AGV. A execução foi conduzida em ambiente industrial real, no pátio de testes da empresa AGVS, localizado em Araucária.

O ambiente de testes consistiu em uma área plana de aproximadamente 6 metros de comprimento por 3 metros de largura, simulando um trecho de operação do AGV dentro da planta. Foram utilizados pallets posicionados em diferentes ângulos e distâncias em relação ao robô, de modo a avaliar a robustez do sistema em condições próximas às de operação real.

Os experimentos foram organizados em três etapas principais:

1. **Aquisição e processamento de dados:** as imagens RGB e de profundidade foram capturadas pela câmera MRDVS S2 Max instalada na parte frontal do AGV, perpendicular ao solo, a uma altura de 55 cm. A execução envolveu a coleta de 350 frames do cenário, armazenados localmente para análise posterior.

2. **Detecção e segmentação:** o modelo YOLOv8n-seg, treinado previamente com o dataset anotado no CVAT, foi utilizado para segmentar os pallets nas imagens RGB. Para cada detecção válida, foram extraídos três pontos de referência: esquerdo, central e direito, com base nas máscaras geradas pelo modelo.

3. **Estimativa de profundidade e projeção da pose:** utilizando a imagem de profundidade correspondente, foram calculadas as distâncias reais dos três pontos em relação à câmera. Em seguida, esses pontos foram transformados para o referencial global por meio das transformações `tf2_ros`, permitindo estimar a pose do pallet e gerar uma trajetória de aproximação para o AGV.

Os experimentos seguiram um protocolo padronizado para garantir comparabilidade entre execuções. No cenário, o AGV foi posicionado a 20 cm do pallet, e a análise foi repetida variando o ângulo do pallet em relação ao eixo frontal do robô (0°, 10°, 20° e 30°). Foram registrados os tempos médios de detecção, as distâncias medidas e os erros de alinhamento calculados a partir das poses estimadas.

Essa estrutura experimental permitiu avaliar não apenas a acurácia da detecção e da profundidade, mas também a estabilidade das transformações espaciais e a confiabilidade do sistema de visão na geração de trajetórias de coleta autônoma.

Tabela 4.1: Configuração de hardware e software utilizada nos testes e nos treinamentos.

Componente	Especificação
AGV	*G*3ET1400V2
Câmera 3D	Lanxin MRDVS S2 (RGB 960×540; Depth 320×240, binning 2×2)
Computador (execução no AGV)	Intel NUC (modelo não identificado via `dmidecode`); CPU Intel Core i7-1195G7 @ 2.90 GHz; RAM 16 GB
Sistema (execução no AGV)	Ubuntu 20.04.6 LTS; ROS 1 Noetic Ninjemys
Treinamento (local, exploratório)	Ultralytics YOLOv8 v8.3.68; PyTorch 2.4.1+cu121; CUDA 12.1; sem GPU dedicada
Treinamento (nuvem, oficial)	Google Colab; Python 3.12.12; Ultralytics 8.3.224; PyTorch 2.8.0+cu126; CUDA 12.6; GPU NVIDIA A100-SXM4-40GB (AMP habilitado)

4.1.1 Ambiente de Execução dos Treinamentos (Colab)

Os treinamentos oficiais do modelo de segmentação foram realizados no Google Colab, com as seguintes características:

- **Plataforma**: Google Colab (Linux base, drivers NVIDIA recentes).
- **Linguagem**: Python 3.12.12.
- **Bibliotecas**: Ultralytics v8.3.224; PyTorch 2.8.0+cu126; CUDA 12.6.
- **Acelerador**: GPU NVIDIA A100-SXM4-40GB, com *Automatic Mixed Precision* (AMP) habilitado.
- **Gerência de configurações**: arquivo `/root/.config/Ultralytics/settings.json` criado automaticamente; visualização via `yolo settings`.

O Colab realizou também validações automáticas de acesso aos dados (latência média de leitura ≈ 0,4 ms) e ajuste do *batch size* para 61% do uso de VRAM, resultando em **batch** de 72 imagens.

4.2 RESULTADOS DO TREINAMENTO DO MODELO YOLOV8-SEG

4.2.1 Hiperparâmetros e Pipeline de Treinamento

O modelo base foi o `yolov8n-seg.pt` (pré-treinado), sobrescrevendo `nc=1` (classe *pallet*). Principais hiperparâmetros efetivos:

- **Tarefa**: segmentação de instâncias (`task=segment`); **imgsz**: 640.
- **Épocas**: 1000; **patience**: 100.
- **Otimizador**: `AdamW` com `lr0=0.002` e `momentum=0.9`; `weight_decay=5e-4`.
- **Batch**: 72 (autoajustado pela VRAM da A100).

- **Augmentations**: `RandAugment`, `fliplr=0.5`, `translate=0.1`, `scale=0.5`, `erasing=0.4`, além de `Blur`/`MedianBlur`, `ToGray` e `CLAHE` (parâmetros padrão do Ultralytics).
- **AMP**: habilitado (verificação e aprovação automática).
- **Arquitetura**: 151 camadas; ≈3,26 M de parâmetros; ≈11,5 GFLOPs; pesos transferidos: 381/417.

O modelo `best.pt`, obtido na época 309 do treinamento oficial, apresentou os resultados resumidos na Tabela 4.2. Os valores indicam alto desempenho tanto em detecção quanto em segmentação, confirmando a consistência do conjunto anotado e da arquitetura YOLOv8n-seg.

Tabela 4.2: Métricas do modelo `best.pt` (treinamento interno na empresa).

Métrica	Precision	Recall	mAP@0.50	mAP@0.50:0.95
Box	0.998	1.000	0.995	0.964
Mask	0.971	0.972	0.977	0.604

O treinamento foi interrompido na época 409 de 1000, com uso máximo de 11,3 GB da GPU A100, indicando boa estabilidade e convergência antes do limite definido. A média de instâncias por imagem no conjunto de validação foi de 1 objeto por cena (36 imagens no total).

4.2.2 Dataset e Divisão

O conjunto final empregado para treino/validação foi composto por 132 imagens segmentadas (classe única). A divisão observada nos logs do Ultralytics foi:

- **Treino**: 96 imagens
- **Validação**: 36 imagens

4.3 SEGMENTAÇÃO, PROFUNDIDADE E TEMPO DE INFERÊNCIA

Para associar cada ponto da máscara (RGB 1920×1080 ou 960×540) ao mapa de profundidade (320×240 com binning 2×2), aplicou-se reamostragem por fatores fixos (1/3 e 4/9, conforme modo de aquisição) e, para lidar com *depth holes*, realizou-se busca local em janela $k \times k$ (tipicamente $k = 5$) selecionando a menor profundidade válida.

A Figura 4.1 apresenta um exemplo de segmentação produzida pelo modelo YOLOv8n-seg.

A Figura 4.2 ilustra os frames principais envolvidos no cálculo da pose do pallet.

A Figura 4.3 mostra os marcadores e transformações associados ao pallet no RViz.

Figura 4.1: Exemplo de segmentação produzida pelo modelo YOLOv8n-seg. (Foto produzida pelo autor.)

4.4 POSE DO PALLET E TRANSFORMAÇÕES ESPACIAIS

$$\theta = \arctan\left(\frac{y_{\text{dir}} - y_{\text{esq}}}{x_{\text{dir}} - x_{\text{esq}}}\right) \tag{4.1}$$

$$\mathbf{p}_{\text{mundo}} = \text{TF}(\mathbf{p}_{\text{camera\_link}}) \tag{4.2}$$

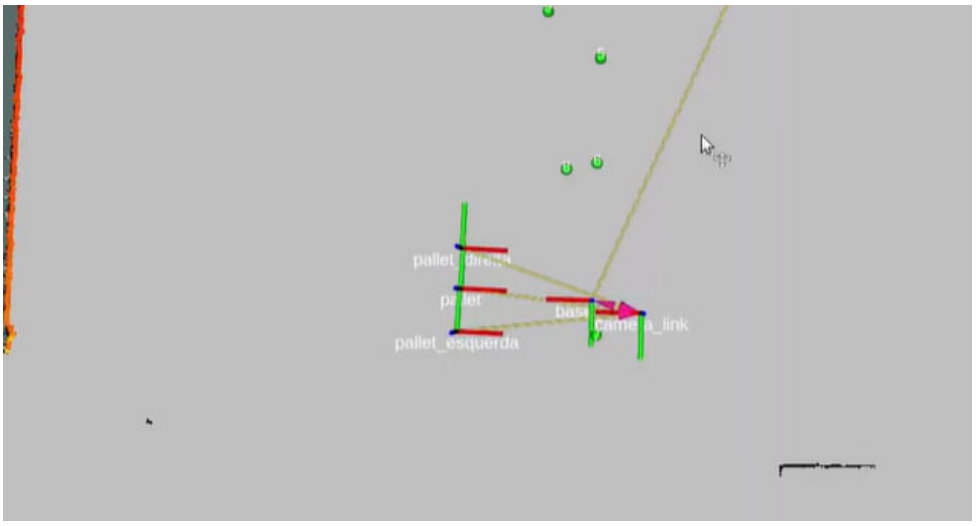


Figura 4.2: Frames principais utilizados no cálculo da pose: visualização das transformações entre camera_link, base_link e mundo. (Foto produzida pelo autor.)

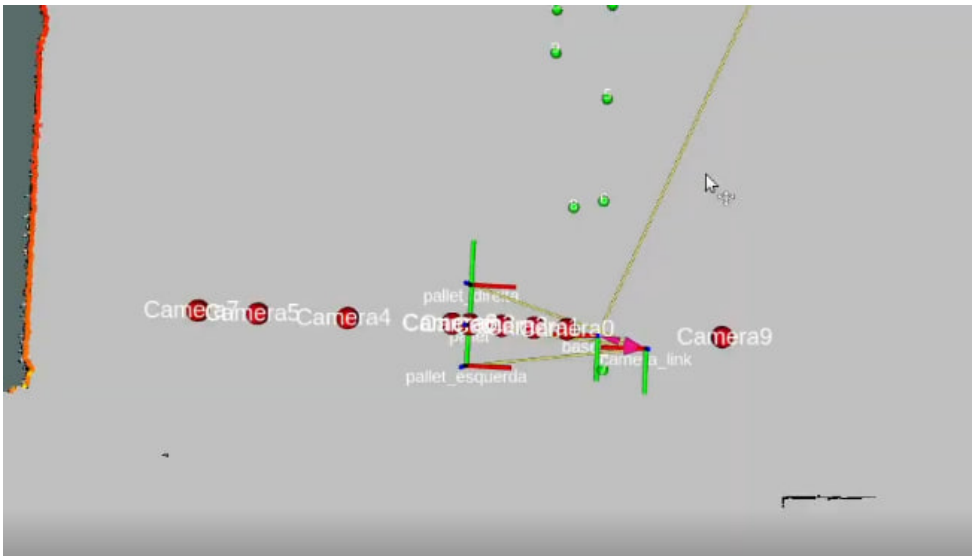


Figura 4.3: Frames principais utilizados no cálculo da pose: visualização dos marcadores e transformações entre camera_link, base_link, pallet e mundo. (Foto produzida pelo autor.)

A Figura 4.4 apresenta o início da manobra de alinhamento realizada pelo AGV.

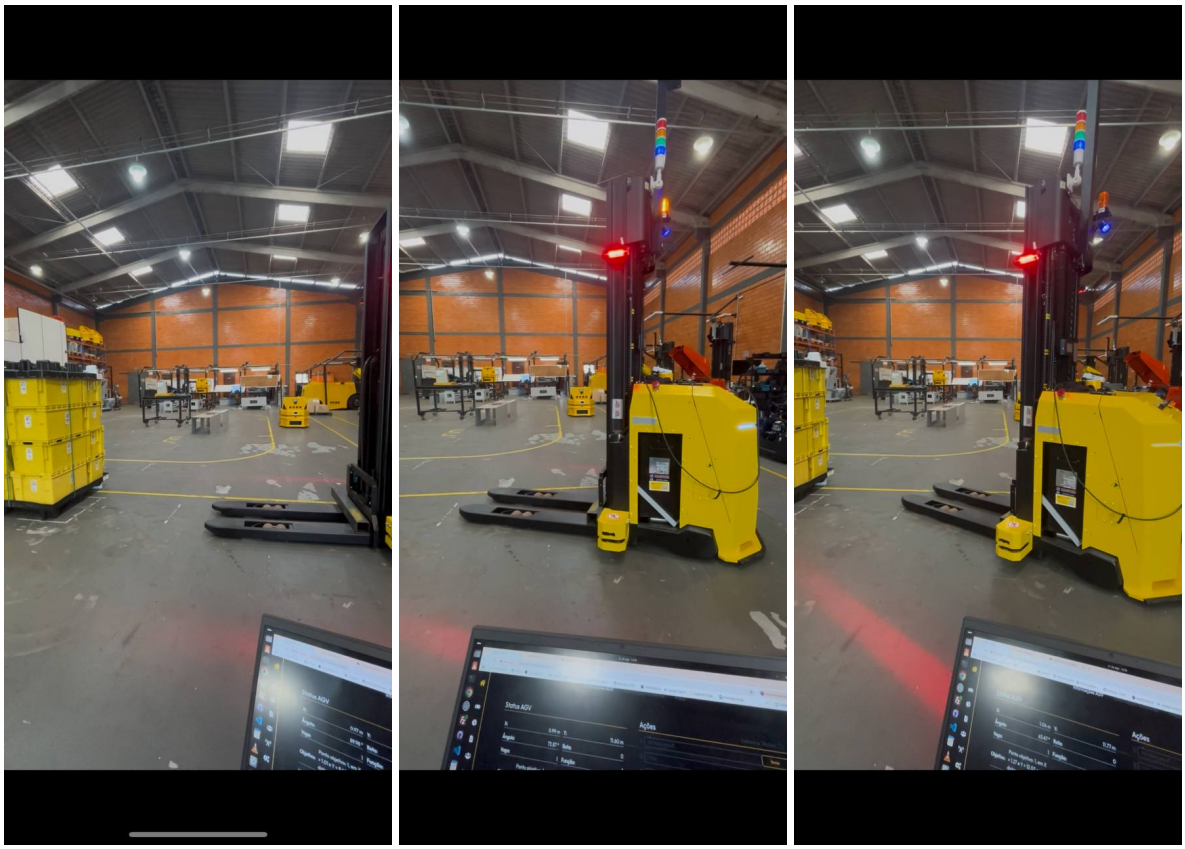


Figura 4.4: Execução da manobra: início do alinhamento demonstrando repetibilidade do sistema. (Foto produzida pelo autor.)

A execução da aproximação e coleta pode ser observada na Figura 4.5.

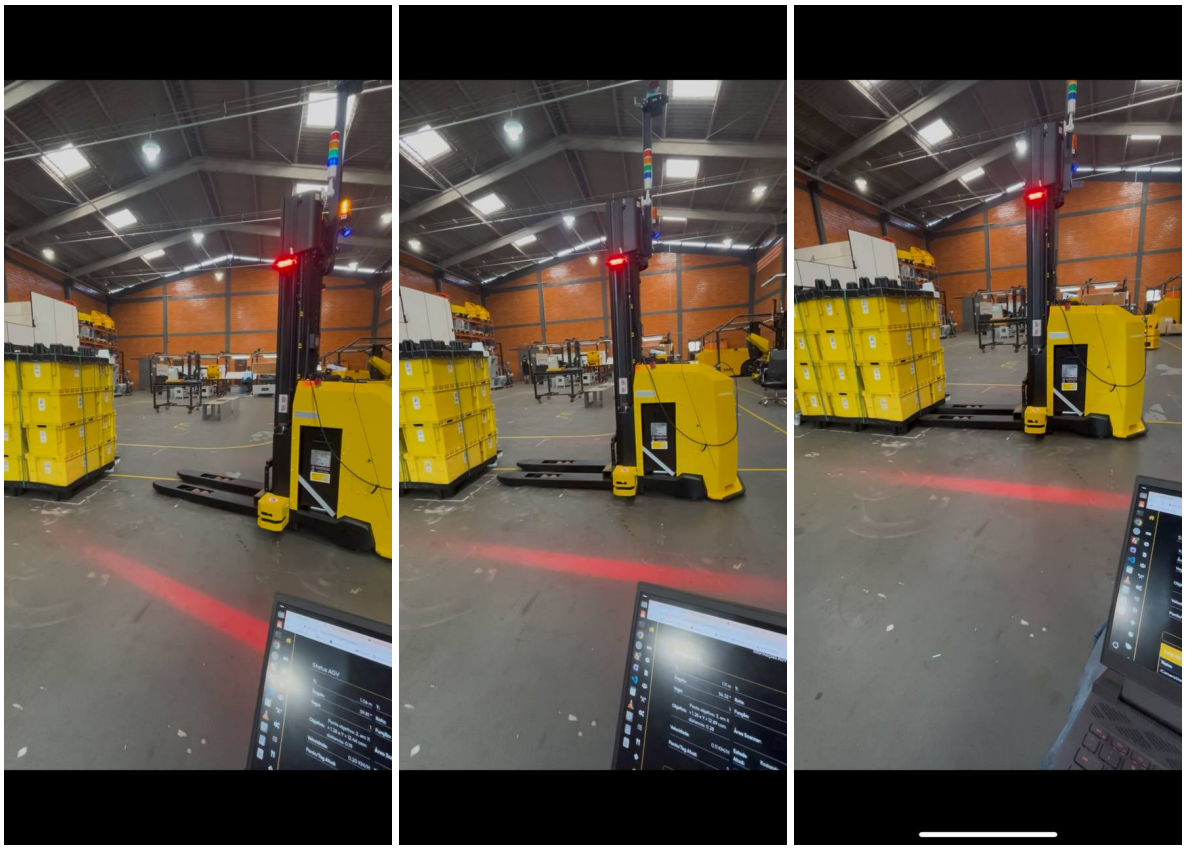


Figura 4.5: Continuação da execução: o AGV completa a aproximação e coleta com sucesso. (Foto produzida pelo autor.)

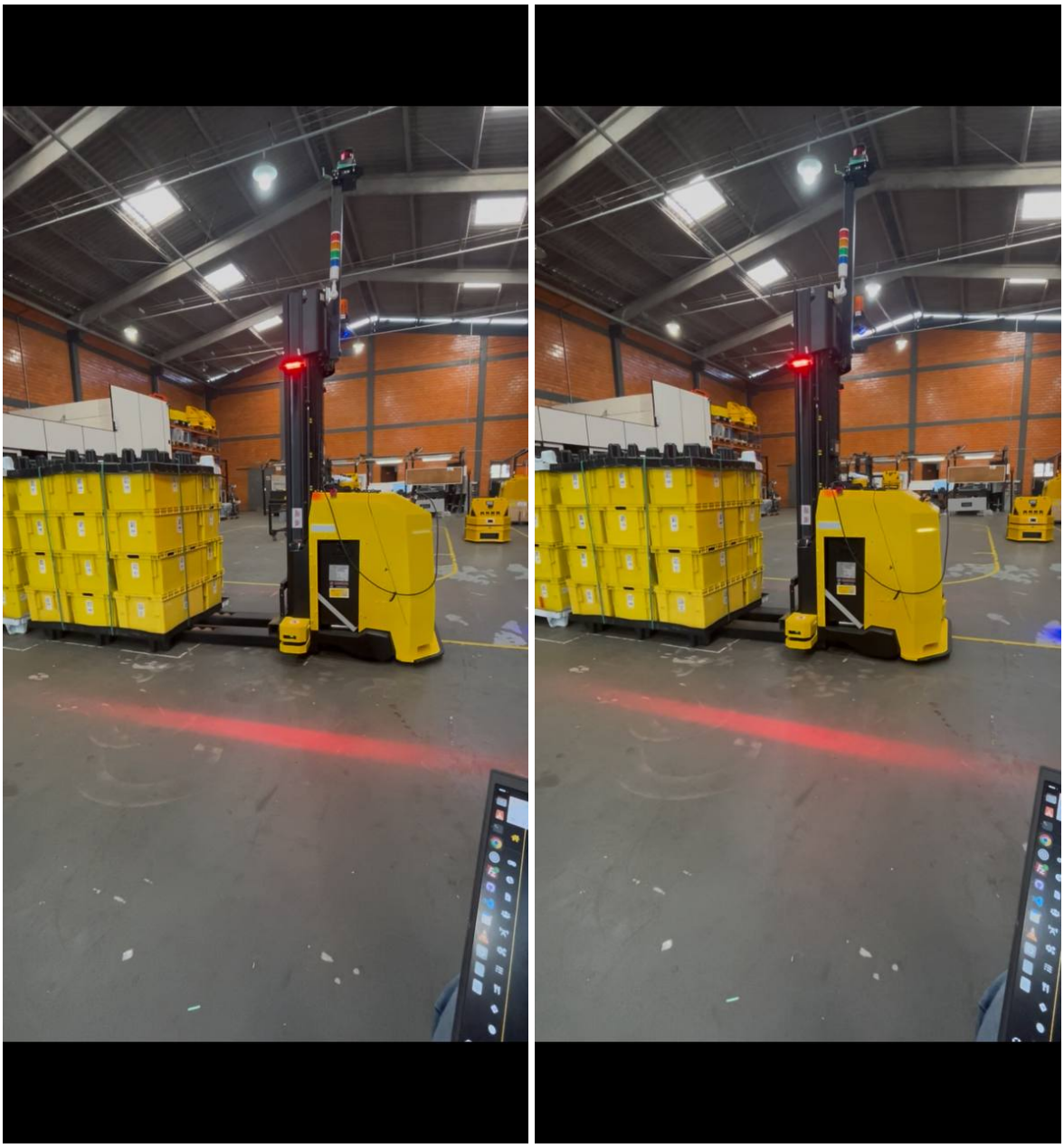


Figura 4.6: Conjunto de frames representativos da execução completa de alinhamento e coleta, evidenciando estabilidade do método. (Foto produzida pelo autor.)

A Figura 4.6 resume a execução completa da manobra de alinhamento e coleta realizada pelo AGV.

De maneira geral, os resultados obtidos demonstram que a solução desenvolvida cumpre integralmente o objetivo central do trabalho: permitir que o AGV identifique com precisão a orientação real do pallet e execute, de forma autônoma, uma trajetória corrigida para sua coleta. A combinação entre segmentação por instâncias, análise robusta de profundidade e uso consistente das transformações espaciais do ROS mostrou-se eficaz e confiável mesmo em condições adversas, como iluminações irregulares, pallets de diferentes materiais e ângulos de desalinhamento de até 30°. Além disso, a estratégia de geração dinâmica dos pontos de navegação provou ser uma abordagem prática e adaptável, reduzindo significativamente a dependência de marcações fixas no chão ou da precisão humana no posicionamento do pallet. Embora limitações permaneçam — incluindo a sensibilidade a oclusões e a necessidade de maior resiliência em

ambientes industriais mais complexos — o desempenho observado evidencia que o pipeline proposto oferece uma base sólida para o aprimoramento da autonomia de AGVs, abrindo caminho para evoluções futuras, como fusão multimodal com LiDAR, modelos de visão de maior robustez ou replanejamento contínuo de trajetória em tempo real.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento, a implementação e a validação de um sistema completo de percepção e navegação para a coleta autônoma de pallets por um AGV, integrando visão computacional baseada em aprendizado profundo, análise de profundidade e transformações espaciais no ROS. Diferentemente de abordagens tradicionais que dependem de posicionamento humano preciso ou de marcações fixas no ambiente, a solução proposta permitiu ao AGV adaptar sua trajetória em tempo real com base na orientação real do pallet, resolvendo um problema recorrente em ambientes industriais: a variabilidade na colocação das cargas.

Os resultados experimentais demonstraram que o modelo de segmentação YOLOv8n-seg, treinado com um conjunto enxuto porém representativo de imagens anotadas no ambiente real, foi capaz de identificar com alta precisão os contornos dos pallets, fornecendo máscaras consistentes mesmo sob variações de iluminação e textura. A integração entre as máscaras segmentadas, o mapa de profundidade da câmera MRDVS e a cadeia de transformações `tf2_ros` possibilitou reconstruir a pose do pallet no referencial global com estabilidade e confiabilidade, garantindo ao AGV informações métricas adequadas para o alinhamento fino.

Além disso, a estratégia de geração dinâmica dos pontos corrigidos de navegação, baseada na pose estimada, mostrou-se eficaz para ajustar o comportamento do robô às condições reais do ambiente, permitindo aproximar-se do pallet de forma suave e orientada mesmo diante de desalinhamentos significativos. Em todos os cenários avaliados, o AGV foi capaz de executar a manobra de coleta com sucesso, evidenciando que o pipeline desenvolvido é robusto e adequado para aplicações industriais.

Embora algumas limitações permaneçam, como a dependência da visibilidade total do pallet e a sensibilidade a reflexos na leitura de profundidade, o desempenho obtido comprova a viabilidade, eficiência e relevância da solução proposta. Assim, o trabalho contribui não apenas como uma solução prática, mas também como uma base tecnológica sólida para o avanço de AGVs mais inteligentes, autônomos e capazes de operar de forma segura e eficiente em ambientes industriais dinâmicos e parcialmente estruturados.

5.1 TRABALHOS FUTUROS

A partir dos resultados obtidos e das limitações observadas, diversas direções promissoras podem ampliar a robustez, a autonomia e o alcance do sistema desenvolvido:

- **Fusão multimodal de sensores (RGB-D + LiDAR + UWB).** A combinação de profundidade com dados de LiDAR 2D/3D e sistemas de localização UWB pode aumentar a precisão espacial do sistema, sobretudo em ambientes com oclusões, forte reflexo no piso ou múltiplos objetos próximos.

- **Extensão para múltiplos pallets simultâneos.** O sistema atual opera com um pallet por cena. Futuras versões podem incluir lógica de multi-target, seleção de prioridade e tomada de decisão em cenários com várias cargas visíveis.

- **Uso de modelos de visão multimodais ou baseados em Transformers.** Arquiteturas mais recentes, como Segment Anything (SAM), RT-DETR ou modelos híbridos YOLO + ViT, podem melhorar a segmentação sob condições extremas, como pallets parcialmente danificados, sujos ou muito inclinados.

- **Replanejamento adaptativo de trajetória em tempo real.** Incorporar algoritmos de replanejamento contínuo, permitindo ao AGV recalcular sua aproximação à medida que novas leituras de visão chegam, aumentando a segurança e a fluidez do movimento.
- **Integração com gêmeos digitais e sistemas de supervisão industrial.** A pose calculada do pallet pode alimentar sistemas de simulação ou ferramentas de MES/WMS, possibilitando monitoramento em tempo real, auditoria de processos logísticos e otimização de fluxos.
- **Treinamento incremental e adaptação contínua.** Implementar aprendizado contínuo (*continual learning*) a partir de novas imagens coletadas durante a operação real, reduzindo degradação de desempenho em cenários não vistos originalmente.
- **Expansão para outras manobras robóticas.** A mesma abordagem pode ser estendida para: – posicionamento para empilhamento, – alinhamento lateral, – inspeção visual de pallets, – detecção de pallets danificados ou carregados.
- **Migração para ROS 2 e middlewares industriais modernos.** A adoção de ROS 2 Jazzy pode melhorar paralelismo, segurança, determinismo e integração com padrões de mercado (DDS, OPC-UA, MQTT), tornando a solução mais escalável para células industriais maiores.
- **Atualização para modelos mais recentes da família YOLO (ex.: YOLOv11)** Uma extensão natural deste trabalho consiste na avaliação e migração para versões mais recentes da família YOLO, como o YOLOv11, que apresentam avanços significativos em velocidade, eficiência e robustez, especialmente para segmentação em ambientes industriais. A adoção desses modelos pode reduzir falsos positivos, melhorar a segmentação das extremidades do pallet e aumentar a estabilidade sob condições desafiadoras, como reflexos no piso ou pallets parcialmente ocluídos.

Essas direções consolidam o potencial do sistema desenvolvido e apontam para uma evolução contínua do AGV rumo a um comportamento mais autônomo, perceptivo e confiável dentro da lógica da Indústria 4.0.

REFERÊNCIAS

Abajo, M. R., Sierra-García, J. E. e Santos, M. (2022). Evolutive tuning optimization of a pid controller for autonomous path-following robot. Em González, H. S., López, I. P., Bringas, P. G., Quintián, H. e Corchado, E., editores, *16th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2021)*, páginas 451–460. Springer International Publishing, Cham.

Åström, K. J. e Murray, R. M. (2008). *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press.

Bochkovskiy, A., Wang, C.-Y. e Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. Acessado em 28/06/2025.

Buda, M., Maki, A. e Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259. Acessado em 28/06/2025.

Campilho, R. D. S. G. e Silva, F. J. G. (2023). Industrial process improvement by automation and robotics. *Machines*, 11(11). Acessado em 28/06/2025.

DATALABELIFY (2023). CVAT2023: Mastering computer vision annotation with a step-by-step guide. `https://www.datalabelify.com`. Acessado em 10/07/2024.

Fragapane, G. et al. (2022). Exploring the impact of industry 4.0 on manufacturing flexibility. *Procedia CIRP*, 81:958–963.

Fraifer, M. A., Coleman, J., Maguire, J., Trslić, P., Dooly, G. e Toal, D. (2024). Autonomous forklifts: State of the art—exploring perception, scanning technologies and functional systems—a comprehensive review. *Electronics*, 14(1):153. Acessado em 28/06/2025.

Golroudbari, A. A. e Sabour, M. H. (2023). Recent advancements in deep learning applications and methods for autonomous navigation: A comprehensive review. Acessado em 28/06/2025.

Hu, X., Luo, Z. e Jiang, W. (2020). Agv localization system based on ultra-wideband and vision guidance. *Electronics*, 9(3). Acessado em 28/06/2025.

Ivanov, D. et al. (2019). A digital supply chain twin for managing the disruption risks and resilience in the era of industry 4.0. *Production Planning & Control*, 30(8):601–614.

Kim, S., Lee, J. e Park, Y. (2019). Tuning PID controller for agv based on industrial environments. *International Journal of Precision Engineering and Manufacturing*, 20(1):91–101.

Lin, Q., Huang, Y., Tan, J. e Xu, B. (2025). Overview of research methods for recognition and localization of logistics pallets based on agv. *ACM International Conference Proceeding Series*. Acessado em 28/06/2025.

Mueller, H., Kim, Y., Gee, T. e Nejati, M. (2025). Pallet detection and localisation from synthetic data. Acessado em 28/06/2025.

Nguyen, D. Q. A., Nguyen, V., Moon, S. e Jo, D. (2023). Pallet detection and pose estimation using fusion of RGB and depth data. Em *2023 20th International Conference on Ubiquitous Robots (UR)*. IEEE. Acessado em 28/06/2025.

Ryck, K. D. (2020). Automated guided vehicles: A survey of techniques and applications. *Journal of Automation and Control*, 58(4):231–240.

Sapkota, R., Ahmed, D. e Karkee, M. (2024). Comparing YOLOv8 and Mask R-CNN for object segmentation in complex orchard environments. https://arxiv.org/abs/2312.07935. Acessado em 28/06/2025.

Schwab, K. (2017). *The Fourth Industrial Revolution*. Crown Business.

TENSORGIRL (2023). Collect and label images to train a YOLOv8 object detection model. https://wandb.ai/tensorgirl/yolov8-collect-and-label/reports/Collect-and-Label-Images-to-Train-a-YOLOv8-Object-Detection-Model--Vmlldzo2MzQ1Nzcw. Acessado em 10/07/2024.

Terven, J., Córdova-Esparza, D.-M. e Romero-González, J.-A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716. Acessado em 28/06/2025.

Ultralytics (2023). YOLOv8 - Ultralytics YOLO Docs. https://docs.ultralytics.com. Acessado em 28/06/2025.

V., V. et al. (2020). Cvat: Computer vision annotation tool. https://github.com/openvinotoolkit/cvat. Acessado em 28/06/2025.

Wang, L., Zhang, Y. e Li, Z. (2017). An adaptive pid control method for autonomous vehicle path tracking. *IEEE Transactions on Industrial Electronics*, 64(5):3893–3902.