

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA AEROESPACIAL

ÍTALO COELHO

UMA APLICAÇÃO DE VISÃO COMPUTACIONAL PARA PADRONIZAÇÃO DE
PROCESSO EM UM ROBÔ DE SERVIÇO ALIMENTAR

Joinville
2024

ÍTALO COELHO

UMA APLICAÇÃO DE VISÃO COMPUTACIONAL PARA PADRONIZAÇÃO DE
PROCESSO EM UM ROBÔ DE SERVIÇO ALIMENTAR

Trabalho apresentado como requisito
para obtenção do título de bacharel
em Engenharia Aeroespacial do
Centro Tecnológico de Joinville da
Universidade Federal de Santa Catarina.

Prof. Dr. Pablo Andretta Jaskowiak

Joinville
2024

ÍTALO COELHO

UMA APLICAÇÃO DE VISÃO COMPUTACIONAL PARA PADRONIZAÇÃO DE
PROCESSO EM UM ROBÔ DE SERVIÇO ALIMENTAR

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de bacharel em Engenharia Aeroespacial, na Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Joinville (SC), 13 de dezembro de 2024.

Banca Examinadora:

Prof. Dr. Pablo Andretta Jaskowiak
Orientador
Presidente

Prof^a. Dra. Anelize Zomkowski Salvi
Membra
Universidade Federal de Santa Catarina

Prof. Dr. Antônio Otaviano Dourado
Membro
Universidade Federal de Santa Catarina

Dedico este trabalho aos meus pais, Adriana e Roberto e à minha irmã Anita.

AGRADECIMENTOS

Em primeiro lugar, agradeço aos meus pais, Adriana e Roberto, cuja dedicação e carinho forneceram, além de todas as oportunidades, a coragem para buscar todos os meus objetivos. Todos os livros que leram para mim antes de dormir e a sabedoria transmitida me incentivaram a buscar o conhecimento para transformar minhas ideias em realidade.

Carinhosamente, agradeço aos amigos que fiz durante a graduação, com quem compartilhei a maior parte desses anos, tanto nos momentos de estudo quanto nos de lazer e descanso. Vocês foram essenciais nessa jornada e a tornaram muito mais agradável.

Agradeço a todos os professores da Universidade de Federal de Santa Catarina - Campus Joinville que, com sua dedicação, compartilharam comigo parte de seus conhecimentos. Em especial, sou grato ao professor Dr. Pablo Andretta Jaskowiak, cujos ensinamentos e persistência fizeram este trabalho possível.

Não poderia deixar de agradecer também à CookingHands Automation e à Roboteria por terem feito parte do início da minha trajetória profissional e por forneceram todos os materiais e ferramentas utilizados para o desenvolvimento deste trabalho.

“Qualquer tecnologia suficientemente avançada é indistinguível da magia.” (Arthur C. Clarke).

RESUMO

A automação e a robótica estão cada vez mais presentes na vida população, a fim de reduzir custos de operação, aumentar a produtividade e realizar tarefas precisas. Com a crescente escassez de mão de obra disponível para o setor de serviços, a automação se apresenta como uma solução para a venda e o preparo de alimentos. Para que seja possível atender os clientes com qualidade, é necessário implementar sistemas de controle capazes de garantir a padronização dos alimentos produzidos por máquinas. Neste trabalho, é apresentado um sistema capaz de dosar o peso de um sorvete produzido por um braço robótico e um sistema de visão computacional capaz de medir a altura e área final do produto. Para isso, foi desenvolvido uma placa de circuito impresso responsável pela leitura do sensor de peso e um programa capaz de identificar o contorno do sorvete e obter suas dimensões. Além disso, foi realizada uma análise dos dados armazenados por uma unidade em operação, com o objetivo de avaliar a possibilidade de se obter uma correlação entre as dimensões do sorvete e o seu peso. Contudo, nenhuma relação entre esses parâmetros foi encontrada e o sistema de medição poderá ser usado exclusivamente para ajustar o valor alvo do peso e aprimorar a consistência do aspecto visual de quantidade do sorvete.

Palavras-chave: Visão Computacional. Automação. Robótica. Padronização. Serviço de Alimentos.

ABSTRACT

Automation and robotics are becoming ever more prevalent in people's day to day lives, as they reduce operational costs, boost productivity, and can perform precise tasks. With the growing scarcity of available labor in the service sector, automation presents itself as a viable solution for food sales and preparation. To ensure high-quality customer service, it is essential to implement control systems capable of standardizing the food produced by machines. This work presents a system capable of dosing the weight of ice cream produced by a robotic arm and a computer vision system designed to measure the final height and area of the product. For this purpose, a printed circuit board was developed to read the weight sensor, along with a program capable of identifying the ice cream's contour and obtaining its measurements. Additionally, an analysis was conducted on data stored by an operational unit in order to evaluate the possibility of establishing a correlation between the ice cream's dimensions and its weight. However, no relationship between these parameters was found, and the measurement system can be used exclusively to adjust the target weight and enhance the consistency of the visually perceived quantity of ice cream.

Keywords: Computer Vision. Automation. Robotics. Standardization. Food Service.

LISTA DE FIGURAS

Figura 1 – Flippy 2 - Estação autônoma de preparo de frituras	17
Figura 2 – Toni - Quiosque de preparo de coquetéis da Makr Shagr	17
Figura 3 – CafeX - Quiosque automatizado de serviço de cafés	18
Figura 4 – Robô de preparo de pizzas da Hyper Robotics	19
Figura 5 – Funcionamento de uma célula de carga	22
Figura 6 – Diagrama funcional do módulo HX711 para leitura de célula de carga	23
Figura 7 – Representação do coeficiente de correlação de Pearson	24
Figura 8 – Representação dos componentes de uma máquina de sorvete <i>soft serve</i>	25
Figura 9 – Quiosque da Roboteria	26
Figura 10 – Processo de dosagem do sorvete por tempo	27
Figura 11 – Sistema de dosagem de sorvete por peso	28
Figura 12 – Câmera utilizada para captura das fotos	29
Figura 13 – Célula de carga utilizada na garra	30
Figura 14 – Vista explodida do conjunto da garra com célula de carga	31
Figura 15 – PCB para leitura da célula de carga	32
Figura 16 – Vista inferior da PCB com a placa de desenvolvimento	32
Figura 17 – Comparação das placas produzidas com componentes THT e SMD	33
Figura 18 – Exemplo da arquitetura de um sistema de distribuição de dados utilizando MQTT	34
Figura 19 – Representação de uma cor no espaço HSV	36
Figura 20 – Representação em escala de cinza dos canais de tonalidade, saturação e intensidade	37
Figura 21 – Comparação entre máscaras com limites diferentes para a cor branca	38
Figura 22 – Resultado da transformação morfológica de fechamento nas máscaras	38
Figura 23 – Contornos obtidos para o limite de cores do sorvete	39
Figura 24 – Exemplos da equalização de histograma e aprimoramento de contraste	40
Figura 25 – Objetos e regiões detectadas pelo modelo <i>Segment Anything</i>	40
Figura 26 – Comparação entre os resultados do SAM e TinySAM	41
Figura 27 – Exemplo de falha na seleção automática de pontos de entrada	42
Figura 28 – Pontos de entrada fixos para o modelo de segmentação contextual .	43
Figura 29 – Dimensões da garra do robô	44
Figura 30 – Distribuição dos valores de altura da garra calculados utilizando a visão computacional tradicional e aprendizado de máquina	45
Figura 31 – Exemplos de imagens rejeitadas para análise	46

Figura 32 – Exemplo de imagem de comparação entre a abordagem tradicional e aprendizado de máquina	47
Figura 33 – Exemplo de resultado obtido com técnicas de processamento	48
Figura 34 – Correlação entre altura do sorvete medida em <i>pixels</i> e o valor calculado em milímetros	49
Figura 35 – Matriz de correlação entre altura, área e peso do sorvete	51
Figura 36 – Matriz de correlação entre área e peso do sorvete com o tempo de dosagem	52
Figura 37 – Sistema de dosagem de sorvete por imagem	53
Figura 38 – Copos com as réplicas de sorvetes	54
Figura 39 – Processo de medição dos corpos de prova com o braço robótico . .	55
Figura 40 – Posição do ponto central da ferramenta do robô	55
Figura 41 – Exemplos de marcadores ArUco	56
Figura 42 – Exemplos de resultados da medição das réplicas de sorvete	57
Figura 43 – Correlação entre a referência e as medidas dos corpos de prova . .	58
Figura 44 – Comparação entre a imagem completa capturada e a região de interesse analisada	59

LISTA DE TABELAS

Tabela 1 – Altura dos corpos de prova	56
Tabela 2 – Comparação entre a altura de referência e a medida pelo programa	57
Tabela 3 – Comparação entre a altura de referência das amostras e a medida corrigida	58

LISTA DE ABREVIATURAS E SIGLAS

ADC	Analog-to-Digital Converter
CI	Circuito Integrado
CLAHE	Contrast Limited Adaptive Histogram Equalization
HSV	Hue, Saturation and Value
IQR	Interquartile Range
MQTT	Message Queuing Telemetry Transport
PCB	Printed Circuit Board
ROI	Region of Interest
SAM	Segment Anything Model
SMD	Surface Mounted Device
TCP	Tool Center Point
THT	Through-Hole Technology

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.1.1	Objetivo Geral	14
1.1.2	Objetivos Específicos	14
1.2	Organização do trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Automação na indústria de alimentos	16
2.1.1	Robôs na indústria de serviço de alimentos	16
2.1.1.1	Flippy - Preparo de frituras	16
2.1.1.2	Makr Shakr - Robô mixologista	17
2.1.1.3	CafeX - Robô barista	18
2.1.1.4	Hyper Robotics - Robô pizzaiolo	19
2.2	Visão computacional	19
2.2.1	Abordagens de visão computacional	20
2.2.2	Transformações morfológicas	20
2.2.3	OpenCV	21
2.2.4	Segmentação contextual de imagem	21
2.3	Conceito de Eletrônica	21
2.3.1	Célula de carga	21
2.3.2	Amplificador e conversor analógico-digital	22
2.4	Análise de dados	23
2.4.1	Matriz e coeficiente de correlação	23
2.5	Sorvete <i>soft serve</i>	24
2.5.1	Máquina de sorvete	25
3	APLICAÇÃO PRÁTICA	26
3.1	Quiosque de sorvete robotizado	26
3.2	Dosagem do sorvete	27
3.2.1	Controle por tempo	27
3.2.2	Controle por peso	28
4	DESENVOLVIMENTO	29
4.1	Desenvolvimento de hardware	29
4.1.1	Sistema de captura de imagens	29
4.1.2	Sistema de medição de peso	29
4.1.2.1	Célula de carga	30

4.1.2.2	Calibração da célula de carga	30
4.1.2.3	Garra com célula de carga	31
4.1.2.4	Design da PCB	31
4.1.2.5	<i>Firmware</i> do microcontrolador	33
4.1.2.6	Propagação dos dados	34
4.1.2.7	Comunicação com o robô	35
4.2	Desenvolvimento de software	35
4.2.1	Base de dados	35
4.2.2	Visão computacional tradicional	35
4.2.2.1	Limites de cor	36
4.2.2.2	Transformações morfológicas	37
4.2.2.3	Seleção do contorno	39
4.2.2.4	Equalização de histograma e realce de contraste	39
4.2.3	Modelo de segmentação contextual	40
4.2.3.1	Seleção dos pontos de entrada	42
4.2.4	Detecção da garra	43
5	RESULTADOS	46
5.1	Análise exploratória dos dados	46
5.1.1	Comparação entre abordagem tradicional e aprendizado de máquina	47
5.1.2	Correlação entre <i>pixels</i> e milímetros	48
5.1.3	Correlação entre medidas e peso do sorvete	50
5.1.4	Correlação entre medidas e o tempo de dosagem do sorvete	50
5.1.5	Sistema de dosagem por imagem	52
5.2	Calibração	53
5.2.1	Medição dos corpos de prova	54
5.2.2	Captura das imagens	56
5.2.3	Medição por imagem dos corpos de prova	57
6	CONCLUSÕES	60
6.1	Trabalhos futuros	61
	REFERÊNCIAS	62
	APÊNDICE A	66
	APÊNDICE B	68

1 INTRODUÇÃO

A automação de processos é um objetivo claro da civilização humana há muito tempo, seja nos processos de manufatura de produtos ou na realização de tarefas cotidianas. Na indústria, esse conceito ganhou força na Primeira Revolução Industrial, com a introdução de teares automatizados e máquinas a vapor, visando a uma produção mais rápida e eficiente de bens de consumo (VIEIRA, 2021). Atualmente, a automação está presente no cotidiano da população, que interage cada vez mais com máquinas de autoatendimento, sistemas de suporte automatizados e até mesmo robôs sorveteiros.

O conceito de Indústria 4.0 promove a interconexão dos sistemas responsáveis pela fabricação de produtos, agregando informações de sensores e estados de equipamentos com o objetivo de aumentar a eficiência e qualidade dos processos. Seguindo essa diretiva, a visão computacional se apresenta como uma ferramenta extremamente valiosa, capaz de fornecer dados sobre o andamento dos processos e qualidade dos produtos finais que antes dependeriam da intervenção ou análise de um colaborador (ALONSO et al., 2019).

Seguindo a tendência da automação de tarefas repetitivas, a Roboteria ¹ lançou no mercado um quiosque de venda de sorvetes produzidos por braço robótico. Além das vantagens inatas da automação, como rapidez, eficiência e disponibilidade de funcionamento ininterrupto, os quiosques chamam a atenção dos clientes pelo fator de novidade, já que a maioria da população ainda não interage rotineiramente com um robô tão complexo (FILHO, 2020).

Durante a operação dos quiosques o maior problema enfrentado foi garantir a consistência do volume de sorvete servido em cada um dos copos. Sorvetes muito pequenos causam descontentamento dos clientes por acreditarem estar recebendo menos produto que o esperado, mesmo com a entrega do peso ofertado. Já os sorvetes muito grandes podem vazar do copo ou não permitir que sejam adicionados os complementos, como caldas e coberturas.

Uma possibilidade direta de controle de quantidade de sorvete servido é o controle por peso, sendo possível garantir que todos os sorvetes entregues apresentem a mesma massa de produto. Contudo, variações nas condições operacionais e ambientais da máquina de sorvete fazem com que a densidade do produto flutue durante o expediente do quiosque e os sorvetes apresentem discrepâncias grandes de volume, influenciando a experiência visual.

Com o objetivo de aprimorar a padronização da quantidade de sorvete servida, propõe-se o desenvolvimento de um sistema capaz de controlar o peso de sorvete

¹ www.roboteria.com.br

servido e um método para a medição da altura e da área do produto final. Neste trabalho, serão apresentados os dispositivos de hardware desenvolvidos para a medição do peso do sorvete e a metodologia utilizada para identificar e quantificar o volume de sorvete servido.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um sistema de visão computacional capaz de medir a altura e a área do sorvete produzido para aprimorar a padronização do aspecto visual dos produtos.

1.1.2 Objetivos Específicos

- Desenvolver um sistema capaz de dosar a quantidade de sorvete a partir de um alvo de peso pré-definido
- Coletar uma base de dados contendo o peso final do sorvete servido e uma foto do produto
- Desenvolver um programa capaz de obter medidas representativas das características físicas visuais do sorvete
- Avaliar se é possível correlacionar as medidas de altura e área do produto com o seu peso
- Avaliar a aplicabilidade do sistema de visão para melhoria na padronização dos sorvetes servidos

1.2 ORGANIZAÇÃO DO TRABALHO

No capítulo de Fundamentação Teórica (Capítulo 2), serão explicados alguns dos conceitos utilizados para o desenvolvimento do projeto e também outros exemplos de empresas que utilizam robótica e automação no setor de serviço de alimentos. Em seguida, no capítulo de Aplicação Prática (Capítulo 3), é apresentado o quiosque de venda de sorvetes produzidos por robô cujo desafio de padronização motivou o desenvolvimento deste trabalho.

O capítulo de Desenvolvimento (Capítulo 4) apresenta os requerimentos e as decisões tomadas para o projeto dos sistemas eletrônicos responsáveis pela leitura dos sensores e aquisição de dados necessários para a dosagem por peso dos sorvetes assim como a programação do sistema de visão computacional que realiza a medição do produto a partir de uma imagem. O capítulo subsequente, Resultados (Capítulo 5), apresenta a performance do sistema de medição, a análise da correlação entre os dados disponíveis e a comparação dos resultados do sistema de visão para corpos

de prova com tamanho conhecido. Finalmente, no Capítulo 6, são apresentadas as considerações finais do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 AUTOMAÇÃO NA INDÚSTRIA DE ALIMENTOS

A automação tem sido amplamente utilizada na fabricação de produtos alimentícios industrializados há muito tempo. Recentemente, sua aplicação, especialmente com o uso de robótica, tem se popularizado no preparo de alimentos no setor comercial, como no preparo de frituras, sorvetes, e cafés (SAVOREAT, 2022).

A automação no ramo alimentício do setor terciário apresenta-se como uma solução viável frente à crescente escassez de mão de obra e ao aumento dos custos operacionais (BRADLEY et al., 2017). No entanto, uma parcela significativa da população ainda acredita que o preparo de alimentos e bebidas deve ser uma atividade exclusivamente humana (IVANOV; WEBSTER, 2023). Portanto, é imperativo garantir que a qualidade dos alimentos produzidos por mecanismos automatizados seja comparável aos produzidos por trabalhadores humanos para solidificar sua aceitação no mercado.

2.1.1 Robôs na indústria de serviço de alimentos

Em resposta a escassez de mão de obra no setor de serviço de alimentos, principalmente nos países desenvolvidos, diversas empresas de automação entraram no ramo de preparo de alimentos, utilizando braços robóticos para manusear utensílios e equipamentos de cozinha criados para operação humana. Até o momento, os robôs têm sido usados principalmente para realizar tarefas simples e repetitivas onde o discernimento de uma pessoa e sua capacidade de ajustar o processo frente a variações no ambiente não são essenciais ao sucesso do preparo do produto (THE ECONOMIST, 2018).

2.1.1.1 Flippy - Preparo de frituras

O sistema de preparo de frituras Flippy, desenvolvido pela Miso Robotics¹, emprega um braço robótico para automatizar a operação de fritadeiras em cozinhas de restaurantes de *fast-food*. Enquanto essa praça seria tradicionalmente operada por duas pessoas, com o uso do robô apenas uma é necessária para produzir e embalar a mesma quantidade de alimentos. Além disso, as fritadeiras são responsáveis por muitos acidentes em cozinhas comerciais e a utilização do robô reduz os riscos para os funcionários (CASTRODALE, 2023).

¹ www.misorobotics.com

Figura 1 – Flippy 2 - Estação autônoma de preparo de frituras



Fonte: Lin (2024).

2.1.1.2 Makr Shkr - Robô mixologista

A empresa Makr Shkr² desenvolve quiosques autônomos onde braços robóticos preparam coquetéis alcoólicos. Além das receitas prontas, o robô é capaz de produzir receitas customizadas, onde o cliente pode escolher entre as diversas opções de ingredientes disponíveis. Contudo, o principal apelo desse produto é o fator de novidade e a experiência do consumidor que pode observar os movimentos meticulosamente coreografados dos braços robóticos no preparo das bebidas (MAKR SHAKR, 2015).

Figura 2 – Toni - Quiosque de preparo de coquetéis da Makr Shkr



Fonte: Vecchi (2022).

² www.makrshkr.com

2.1.1.3 CafeX - Robô barista

O robô barista da CafeX³ tem como principal objetivo reduzir as filas encontradas pelos clientes ao comprar cafés em locais com alto fluxo de pessoas, como centros comerciais e aeroportos. Especialmente relevante para operação em aeroportos, o quiosque robótico pode operar continuamente 24 horas por dia sem a necessidade de um operador, sendo possível também atender clientes em horários de baixo movimento quando não seria econômico manter funcionários trabalhando (GAVLAK, 2019).

A utilização de um robô elimina a necessidade de treinar baristas para produzir os produtos cada vez mais complexos preferidos pelos clientes das cafeterias. O robô também apresenta, como vantagem para o consumidor, a higiene do processo, livre de contaminações cruzadas acidentais, consistência do produto servido já que o processo é repetido exatamente para todos os pedidos e até mesmo a redução dos preços praticados em regiões onde a mão de obra é custosa.

Figura 3 – CafeX - Quiosque automatizado de serviço de cafés



Fonte: Blum (2019).

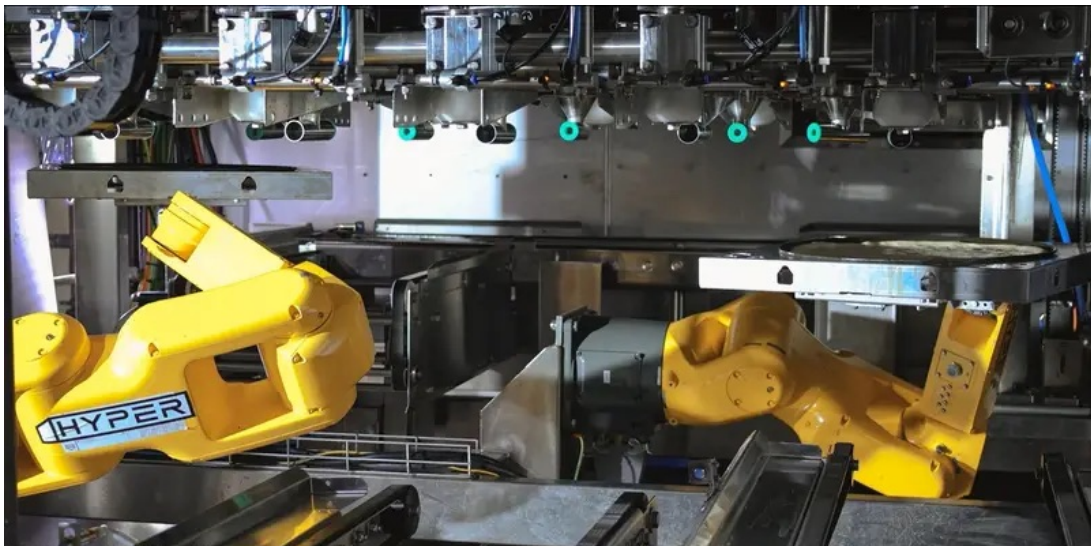
³ www.cafexapp.com

2.1.1.4 Hyper Robotics - Robô pizzaiolo

A empresa Hyper Robotics⁴ possui cozinhas de preparo de pizza em contêineres totalmente automatizados. Os robôs são responsáveis pela produção e embalagem do produto e a estrutura possui um sistema de limpeza de ozônio e luz ultravioleta que higieniza a cozinha sem a intervenção de nenhum humano.

Um grande diferencial dessa unidade é a capacidade de realizar todos os processos, desde a retirada das matérias-primas dos freezers, esticar e trabalhar a massa até o corte das pizzas em fatias. A primeira unidade da pizzaria autônoma é capaz de produzir até 50 pizzas por hora, podendo funcionar em qualquer horário (DUFFY, 2022).

Figura 4 – Robô de preparo de pizzas da Hyper Robotics



Fonte: Hyper Robotics (2024).

2.2 VISÃO COMPUTACIONAL

Um sistema de visão computacional tem como objetivo a extração por uma máquina de informações de interesse presentes em uma imagem. Essa habilidade é essencial para que um sistema automatizado possa adquirir dados sobre o mundo físico sem a necessidade da intervenção de um operador humano. No cotidiano, a visão computacional está presente em diversas áreas como, por exemplo, o reconhecimento de caracteres, utilizado por equipamentos de fiscalização de trânsito, e o reconhecimento facial, em sistemas de controle de acesso e aparelhos eletrônicos de uso pessoal (PETERS, 2017).

⁴ www.hyper-robotics.com

As primeiras tentativas de extrair dados a partir de uma imagem com um computador se iniciaram na década de 1950, notoriamente, com o desenvolvimento do *Perceptron*, uma máquina analógica treinada para reconhecer formas geométricas. Esse dispositivo foi construído a partir de um modelo probabilístico inspirado no funcionamento do sistema nervoso humano, onde a informação é armazenada em conexões e associações que representam a preferência para uma resposta específica (ROSENBLATT, 1958).

Nas décadas subsequentes, com o advento e popularização dos computadores e câmeras digitais, foram desenvolvidos inúmeros algoritmos com o objetivo de extrair características e detectar objetos em imagens. A partir da década de 2000, os esforços foram concentrados na utilização das redes neurais convolucionais para as tarefas de classificação de imagens e detecção de objetos (IBRAHIM, 2024).

2.2.1 Abordagens de visão computacional

Os sistemas de visão computacional podem ser classificados em dois tipos de abordagem: análise descritiva, na qual o desenvolvedor elabora um modelo que representa o processo da aplicação, e análise preditiva, que envolve o uso de aprendizado de máquina, onde um modelo é treinado com um conjunto de dados de entrada, cujas saídas desejadas são previamente conhecidas.

Nas abordagens baseadas em aprendizado de máquina, o modelo é treinado em vez de programado explicitamente, o que reduz a necessidade de conhecimento profundo sobre os detalhes específicos da aplicação. Em contrapartida, os algoritmos de visão computacional tradicionais exigem uma análise detalhada de todos os aspectos e possíveis variações da aplicação, tornando-os menos flexíveis que os métodos baseados em aprendizado de máquina (MAHONY et al., 2019).

2.2.2 Transformações morfológicas

As transformações morfológicas constituem um conjunto de técnicas amplamente utilizadas em sistemas de visão computacional para melhorar a demarcação de regiões ou objetos de interesse. Essas transformações são aplicadas em imagens monocromáticas ou em escala de cinza, analisando cada pixel e sua vizinhança imediata com base em um elemento estrutural, que é definido conforme os requisitos da aplicação (ROSEBROCK, 2021b).

As transformações morfológicas básicas são a dilatação, que tem como efeito aumentar o tamanho do sujeito de interesse, e a erosão, que o reduz. A dilatação é utilizada quando é necessário fechar buracos na região de seleção ou unir dois sujeitos distintos que representam a mesma região de interesse. Já a erosão, por ser um processo inverso a dilatação, é aplicada para remover detalhes e rugosidades

indesejados na região de contorno do objeto ou separar uma região de interesse em dois sujeitos.

2.2.3 OpenCV

O OpenCV (BRADSKI, 2000) é um projeto de código aberto amplamente utilizado na indústria, disponível como bibliotecas em Python e C++. Seus principais atrativos são sua versatilidade, robustez, facilidade de uso e disponibilidade de materiais de referência. Por ser de código aberto, pode ser modificado para atender as necessidades de um projeto, de forma gratuita, pelo próprio usuário e desenvolvedores do mundo todo são encorajados a oferecer aprimoramentos e soluções para problemas encontrados na sua utilização.

2.2.4 Segmentação contextual de imagem

A segmentação contextual de imagens diz respeito à tarefa de agrupar os *pixels* de uma imagem correspondente a um mesmo objeto ou a uma parte específica de um objeto. Com a introdução das redes neurais nesse processo, a precisão aumentou consideravelmente, o que ampliou sua utilidade em diversas aplicações. Atualmente, a segmentação de imagem com auxílio de redes neurais é amplamente aplicada em sistemas de produção e controle de qualidade automatizados, sistemas de assistência de direção avançada e detecção de anomalias médicas em exames de imagem (LIU; DENG; YANG, 2018).

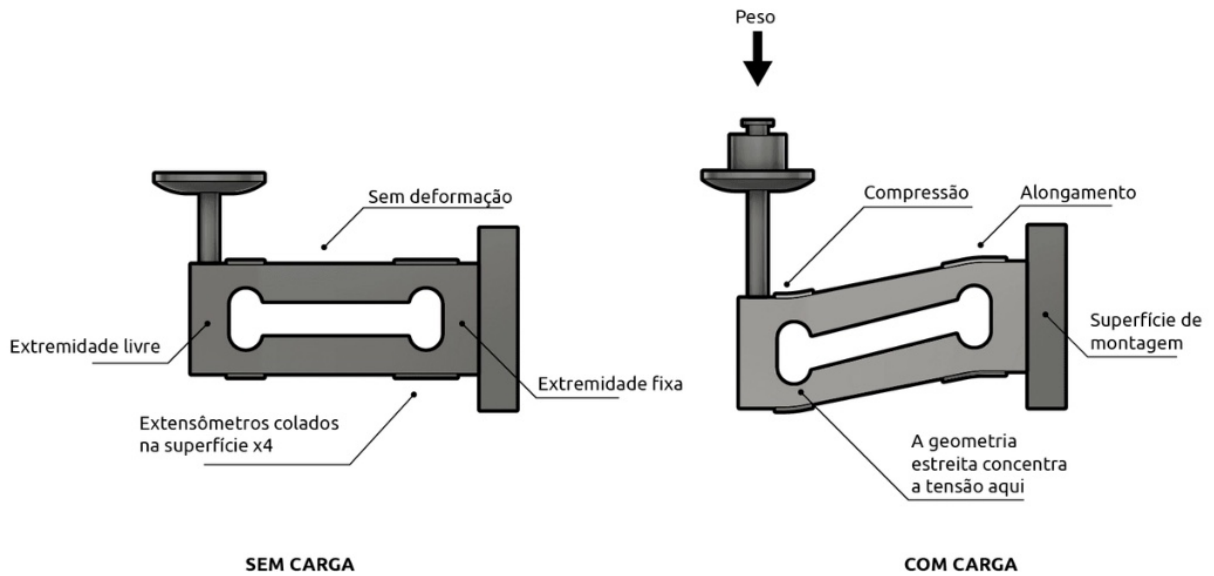
2.3 CONCEITO DE ELETRÔNICA

Para o desenvolvimento deste trabalho, foi necessário compreender alguns conceitos de eletrônica referentes aos sensores e componentes utilizados. A seguir, são apresentados os princípios de funcionamento do sensor de peso e do módulo responsável pela sua leitura.

2.3.1 Célula de carga

As células de carga são transdutores que convertem a força aplicada sobre elas em um sinal elétrico proporcional. O tipo mais comum, utilizado neste trabalho, é baseado em uma ponte de Wheatstone com extensômetros (sensores cuja resistência elétrica varia conforme são submetidos a compressão ou alongamento). Para possibilitar a medição da força, as células de carga são fabricadas a partir de um bloco sólido de metal, com uma região aliviada onde os extensômetros são posicionados, concentrando a deformação mecânica nessa área específica (FLINTEC, 2024).

Figura 5 – Funcionamento de uma célula de carga



Fonte: Flintec (2024).

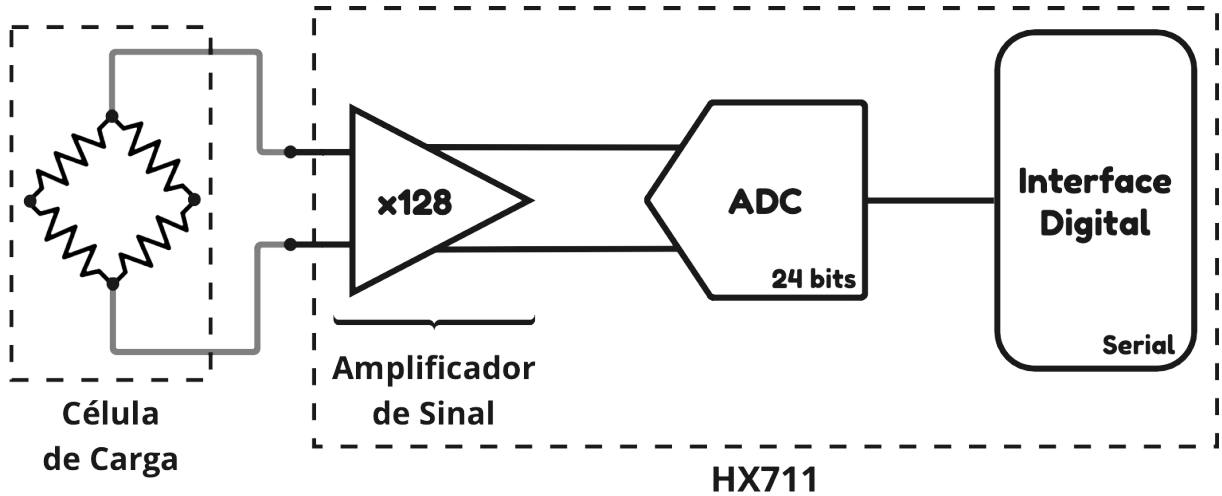
Como pode ser visto na Figura 5, ao posicionar um peso sobre uma das extremidades da célula de carga mantendo a outra fixa, dois extensômetros sofrem compressão e dois sofrem alongamento. A variação nas dimensões dos sensores causam um desbalanceamento nos valores das resistências da ponte Wheatstone proporcional a carga aplicada.

2.3.2 Amplificador e conversor analógico-digital

Para que o sinal de tensão gerado por uma célula de carga possa ser processado por um dispositivo de controle, é necessário amplificá-lo e convertê-lo em sinais digitais compatíveis com um microcontrolador. A amplificação é necessária porque a tensão do sinal fornecido pelas células de carga é muito baixa, na ordem de milivolts, e precisa ser elevada a níveis detectáveis pelo circuito de conversão analógico-digital (ADC) (CASTRO, 2021). Esse conversor, por sua vez, transforma o sinal analógico em uma sequência de dados binários, permitindo que o microcontrolador processe as leituras das forças aplicadas.

A Figura 6 apresenta o diagrama funcional simplificado de um CI comumente utilizado para interface entre uma célula de carga e sistema de controle digital. Esse módulo possui um amplificador para os sinais de tensão recebidos pela célula de carga, um conversor analógico-digital e uma interface serial para acesso das medições de tensão.

Figura 6 – Diagrama funcional do módulo HX711 para leitura de célula de carga



Fonte: Autor (2024).

2.4 ANÁLISE DE DADOS

A análise de dados é fundamental para extrair informações relevantes e identificar padrões em um conjunto de informações. No tópico a seguir, serão discutidos conceitos essenciais para as análises realizadas neste trabalho.

2.4.1 Matriz e coeficiente de correlação

A correlação é a medida estatística que descreve a intensidade e a direção da relação entre duas variáveis. Quando as variáveis são correlacionadas, a mudança no valor de uma delas é associada a uma mudança previsível no valor da outra (THEVAPALAN, 2024). O coeficiente de correlação mais comumente utilizado é o coeficiente de Pearson, que pode ser calculado para as variáveis x e y conforme a Equação (1).

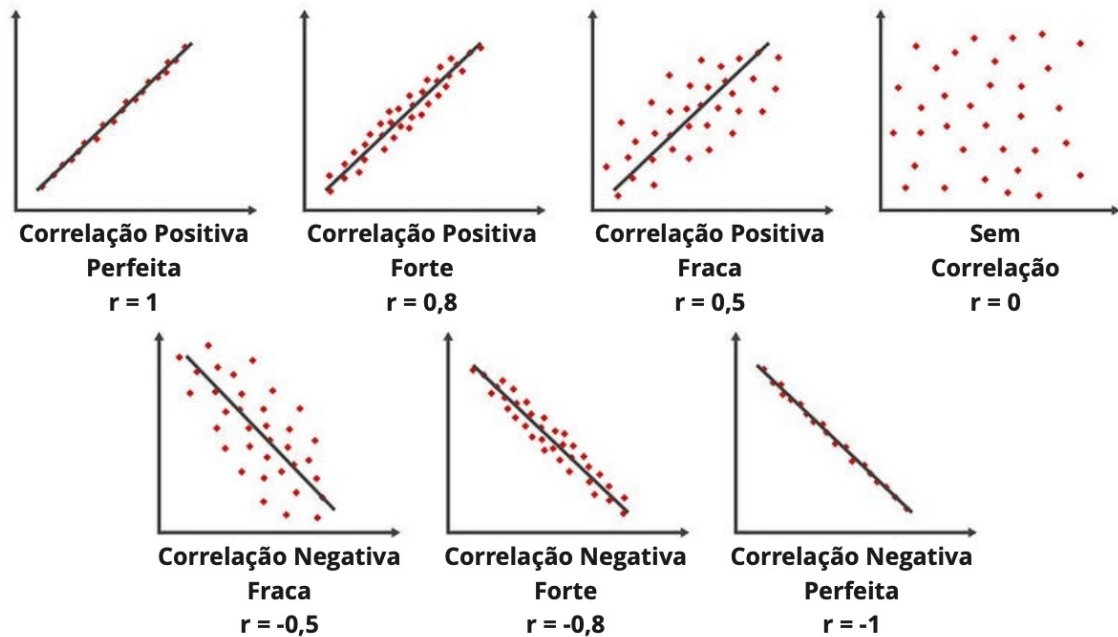
$$r(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

Na Equação (1), x_i e y_i representam os valores individuais das variáveis analisadas enquanto \bar{x} e \bar{y} correspondem a média dos valores desses parâmetros no conjunto de dados analisado.

Um coeficiente de Pearson igual a $+1$ representa uma relação perfeitamente linear entre duas variáveis, onde o aumento no valor da primeira variável faz com que o valor da segunda também aumente. Já no coeficiente com valor igual a -1 , a relação é inversa, ou seja, a redução no valor de um dos parâmetros corresponde a um incremento linear no valor da outra. Caso o coeficiente seja nulo, nenhuma relação

linear pode ser observada entre as variáveis. A Figura 7 apresenta uma interpretação gráfica da correlação, ilustrando o comportamento do coeficiente de Pearson em um conjunto de pontos, destacados em vermelho, que correspondem aos valores das variáveis representadas nos eixos.

Figura 7 – Representação do coeficiente de correlação de Pearson



Fonte: Adaptado de Thevapalan (2024)

Uma matriz de correlação apresenta uma tabela onde cada célula apresenta o coeficiente representativo da relação entre a variável da sua linha e coluna. Dessa forma, é possível identificar padrões presentes em grandes base de dados com múltiplos parâmetros. Essa técnica é amplamente utilizada antes do desenvolvimento de modelos de aprendizado de máquina a fim de determinar quais as variáveis mais relevantes (WAGAVKAR, 2023).

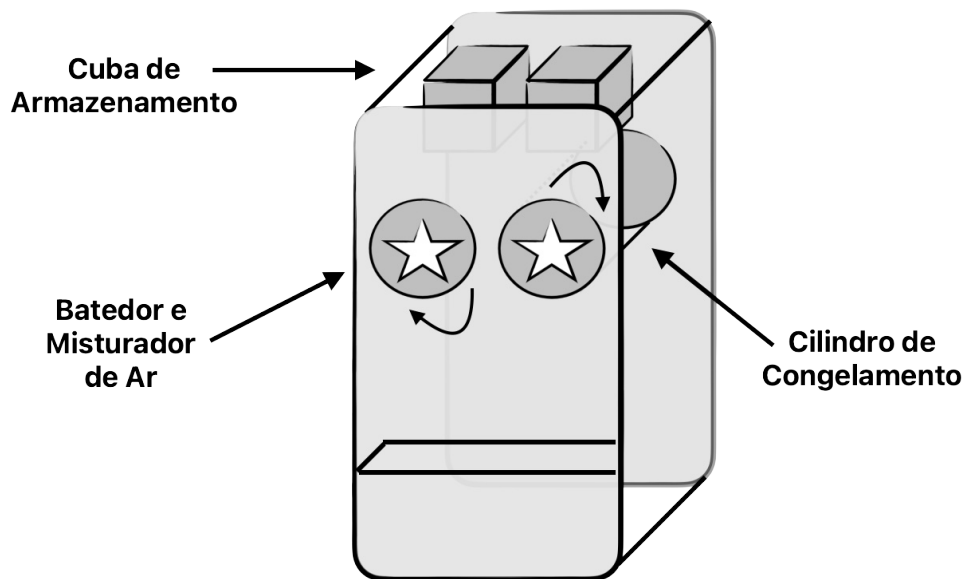
2.5 SORVETE *SOFT SERVE*

O sorvete referenciado neste trabalho é do tipo *soft serve*, popularmente conhecido no Brasil como sorvete italiano. Esse tipo de sorvete, juntamente com a máquina necessária para o seu preparo, foi inventado em 1926 (HARTEL; RANKIN; JR., 2017). Esse produto se diferencia dos sorvetes convencionais por ser produzido apenas momentos antes de ser servido ao cliente. A temperatura do produto é maior do que o sorvete convencional, vendido em supermercados, o que contribui para sua textura mais cremosa.

2.5.1 Máquina de sorvete

Para que o sorvete atinja a sua consistência e sabor característicos, é essencial que a matéria-prima, chamada de *mix*, seja misturada com ar e congelada muito rapidamente para garantir que os cristais de gelo formados no congelamento sejam pequenos o suficiente para passarem despercebidos pelo paladar. As máquinas possuem como parâmetro de controle da característica do produto a viscosidade do sorvete, medido e parametrizado a partir da corrente utilizada pelo motor responsável por misturar o sorvete no cilindro de congelamento (VARGAS, 2023).

Figura 8 – Representação dos componentes de uma máquina de sorvete *soft serve*



Fonte: Autor (2024)

Como mostrado na Figura 8, o *mix* é preservado resfriado na cuba de armazenamento e desce por gravidade para o cilindro de congelamento conforme os sorvetes são servidos e esse é esvaziado. O batedor dentro do cilindro incorpora ar na massa enquanto essa é congelada e misturada até que a viscosidade desejada para o produto seja atingida. Quando a alavanca responsável pela abertura da válvula de saída de sorvete é acionada, o batedor gira expelindo o produto pela abertura frontal. Assim que o produto é servido, a máquina aciona o compressor de refrigeração e o batedor até que os parâmetros configurados sejam atingidos pela mistura de produto pronto e matéria-prima adicionada.

3 APLICAÇÃO PRÁTICA

3.1 QUIOSQUE DE SORVETE ROBOTIZADO

Este trabalho foi desenvolvido com o objetivo de aprimorar a padronização do tamanho dos sorvetes servidos pelos quiosques da Roboteria, que utilizam um braço robótico industrial para preparar sorvetes do tipo *sundae* de forma autônoma. Os pedidos são feitos pelos clientes em telas de autoatendimento, que transferem a receita do produto escolhido para a fila de produção do robô.

Os consumidores tem a opção de escolher entre dois sabores de sorvete distintos, baunilha e chocolate, três opções de caldas líquidas e quatro de complementos sólidos. Os produtos prontos são entregues pelo robô aos clientes através de um porta eclusa, a fim de evitar que as pessoas possam sofrer ferimentos ou contaminar a garra do robô.

A Figura 9 mostra o totem de autoatendimento, onde os pedidos são realizados, ao lado do quiosque, no qual o braço robótico prepara os sorvetes e entrega o produto pronto aos clientes pelas portas laterais.

Figura 9 – Quiosque da Roboteria



Fonte: Arquivo Roboteria (2022).

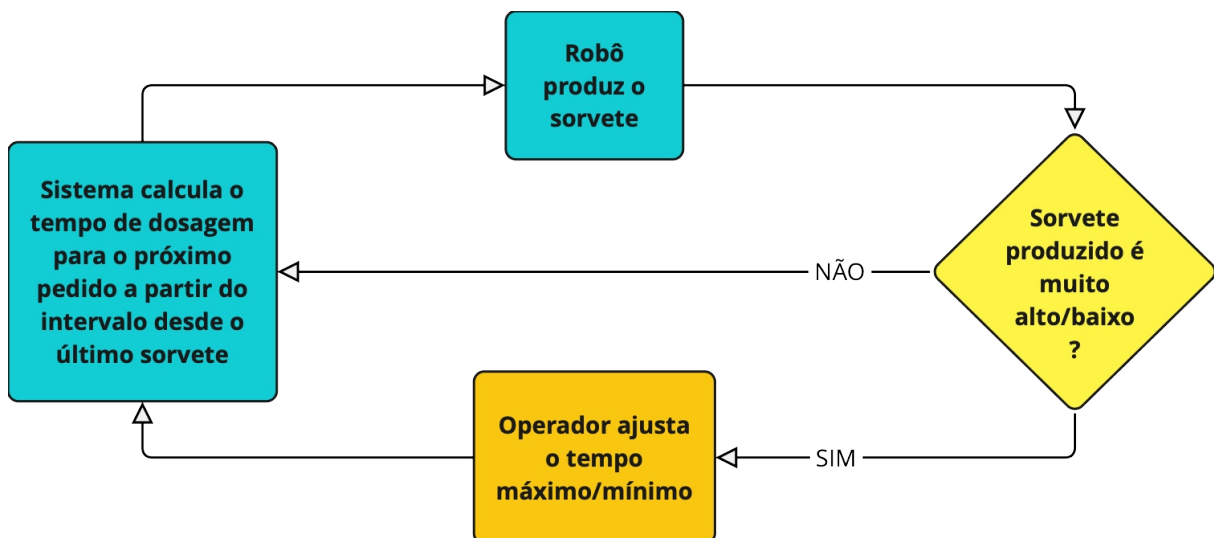
3.2 DOSAGEM DO SORVETE

A máquina responsável pela produção do sorvete foi projetada para ser operada por um humano, sem controle ou medição da quantidade de sorvete dispensado, uma vez que um operador pode facilmente observar visualmente quando o sorvete atinge o tamanho desejado. Dessa forma, foi necessário desenvolver estratégias para o controle da dose do produto servida pelo robô em cada copo.

3.2.1 Controle por tempo

No início das operações dos quiosques, foi implementado um sistema de controle baseado no tempo de abertura da válvula da máquina de sorvete. Esse algoritmo considerava o intervalo desde o último sorvete servido e exigia que um operador, observando o aspecto visual do sorvete, ajustasse os limites máximos e mínimos para o tempo de dosagem. O fluxograma que representa esse processo pode ser visto na Figura 10.

Figura 10 – Processo de dosagem do sorvete por tempo



Fonte: Autor (2024).

A princípio esse método foi adotado pela baixa complexidade de implementação e por não requisitar o desenvolvimento de nenhum dispositivo adicional. Dessa forma, foi possível colocar o quiosque em funcionamento em um momento onde os esforços de engenharia estavam ocupados com o desenvolvimento de outros sistemas.

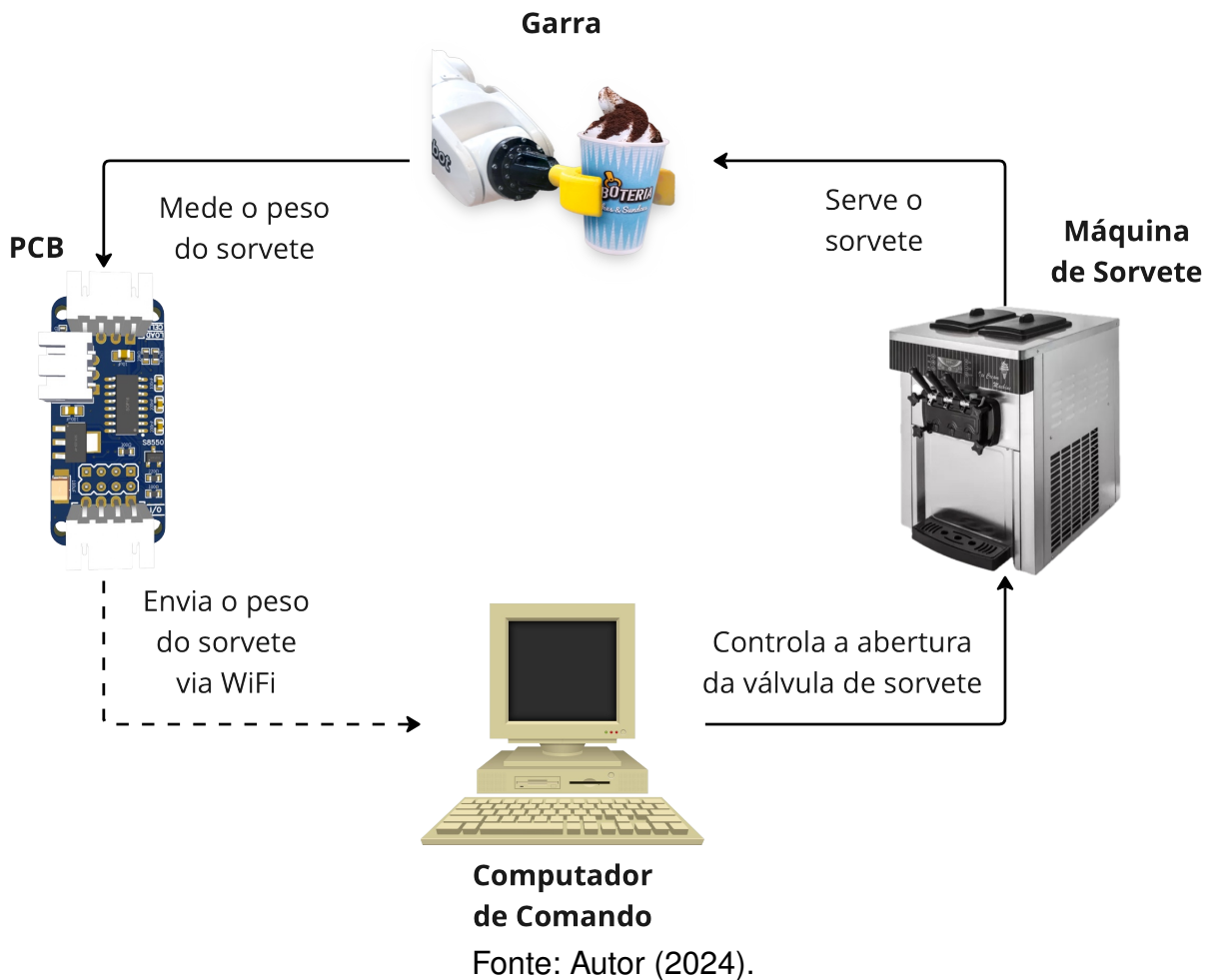
As principais desvantagens desse método são a produção inicial de pelo menos um sorvete fora do padrão desejado antes que o ajuste possa ser realizado, além da dependência de um operador humano para esses ajustes, o que contraria o objetivo de alcançar autonomia total no processo de produção.

3.2.2 Controle por peso

Atualmente, é utilizado um sistema de pesagem que comanda a abertura da válvula responsável pela liberação do sorvete até que o peso desejado seja atingido. Apesar de poder operar sem a necessidade de um operador humano, esse sistema não consegue levar em consideração as variações na densidade do sorvete, que alteram consideravelmente o aspecto visual de altura e volume, percebidos pelos clientes como a quantidade de sorvete servida.

A Figura 11 mostra o funcionamento desse sistema de dosagem, onde a garra envia o peso atual para o computador de comando. Quando o peso desejado é atingido, o computador comanda o fechamento da válvula de sorvete e indica para o robô que esse pode avançar para as próximas etapas no processo de produção.

Figura 11 – Sistema de dosagem de sorvete por peso



4 DESENVOLVIMENTO

4.1 DESENVOLVIMENTO DE HARDWARE

4.1.1 Sistema de captura de imagens

As imagens dos sorvetes utilizadas no desenvolvimento deste trabalho foram capturadas utilizando uma câmera posicionada na lateral da máquina de sorvete. O robô foi programado para se colocar em frente à câmera após terminar de servir o sorvete no copo e avisar o computador de comando que uma foto deve ser capturada a partir do *feed* de transmissão ao vivo da câmera e enviada para a base de dados.

Uma câmera de segurança conectada à rede local do quiosque, como a da Figura 12, foi escolhida pois nenhuma das unidades em operação foi projetada ou construída com a previsão de obter imagens dos sorvetes produzidos. Esses dispositivos são projetados com operação ininterrupta, design compacto e simplicidade de instalação em mente, o que os tornam práticos para utilização em testes nas unidades já em funcionamento.

Figura 12 – Câmera utilizada para captura das fotos



Fonte: Intelbras (2024).

4.1.2 Sistema de medição de peso

Para que o peso dos sorvetes possa ser utilizado no controle da quantidade servida e armazenado para as análises presentes neste trabalho, foi necessário um circuito eletrônico responsável pela leitura do sensor de peso. Além disso, um sistema de aquisição e transmissão de dados precisou ser implementado, permitindo que as

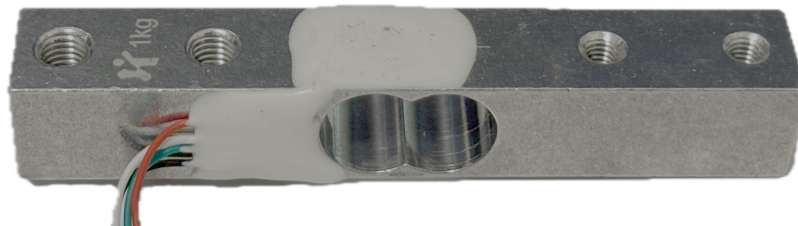
leituras sejam entregues aos sistemas relevantes dentro do processo automatizado.

4.1.2.1 Célula de carga

A resolução da leitura do peso é inversamente proporcional à capacidade de carga do sensor, portanto, utilizar o menor valor de carga possível é essencial para obter melhores resultados. A seleção da capacidade de peso máximo da célula de carga foi feita utilizando um sensor de teste com capacidade de $10kg$ e monitorando o valor máximo lido durante um dia de operação típica do braço robótico. Com base nessa análise, foi possível determinar que o modelo com capacidade de carga de $5kg$ é o mais adequado, por ser o menor valor comercialmente disponível capaz de suportar a força máxima enfrentada pelo manipulador durante a operação.

Foi escolhida uma célula de carga do tipo barra, como a mostrada na Figura 13, com dois pontos de fixação nas extremidades. Esse modelo é o mais indicado para aplicações onde a força aplicada é perpendicular ao sensor.

Figura 13 – Célula de carga utilizada na garra



Fonte: Autor (2024).

4.1.2.2 Calibração da célula de carga

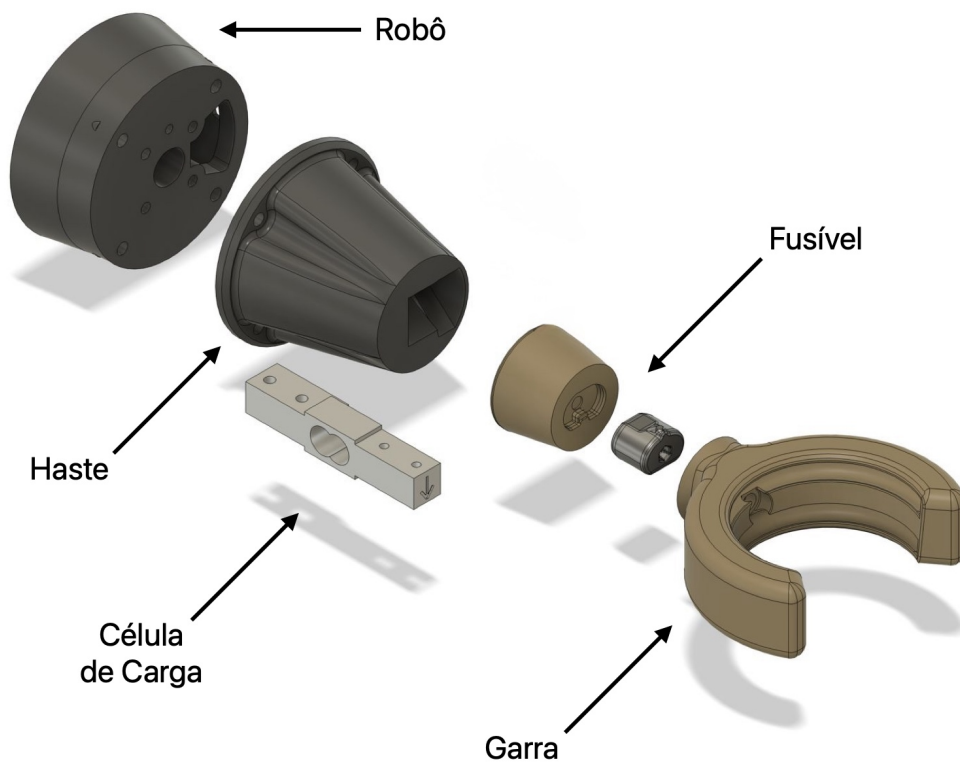
Cada célula de carga apresenta uma saída de tensão ligeiramente diferente para um determinado peso devido às variações no processo de fabricação. Dessa forma, foi necessário implementar uma rotina de calibração no *firmware* do microcontrolador capaz de calcular o fator de calibração do sensor, responsável pela conversão da leitura de tensão em informação de carregamento. Esse valor, f , pode ser calculado através da Equação (2) a partir da variação da tensão medida pelo conversor ADC (ΔV_{ADC}) após a aplicação de uma carga conhecida no sensor ($W_{conhecido}$). Para atingir melhores resultados de medição, é preferível que o peso conhecido aplicado seja próximo ao valor médio do intervalo de interesse de medição (MACHADO et al., 2019).

$$f = \frac{W_{conhecido}}{\Delta V_{ADC}} \quad (2)$$

4.1.2.3 Garra com célula de carga

Para medir o peso dos sorvetes de forma adequada, foi desenvolvida uma garra para o robô composta por três componentes principais, além da célula de carga, apresentados na Figura 14. A haste conecta diretamente a flange do manipulador a uma das extremidades do sensor. O fusível mecânico, ligado à outra extremidade da célula de carga, protege os demais componentes do quiosque contra impactos. Por fim, a garra propriamente dita, conectada ao fusível, é responsável por segurar o copo de sorvete.

Figura 14 – Vista explodida do conjunto da garra com célula de carga



Fonte: Autor (2024).

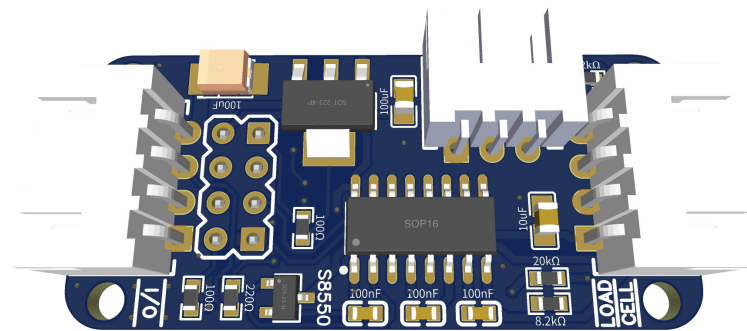
4.1.2.4 Design da PCB

A leitura dos sinais provenientes da célula de carga exige amplificação e conversão para um formato digital que possa ser lido por um microcontrolador. Para isso, foi utilizado o módulo HX711 (AVIA SEMICONDUCTOR, 2024), especificamente desenvolvido para realizar a interface de células de carga com microcontroladores. Esse módulo possibilita a leitura dos sinais com uma resolução de 24 bits e uma frequência máxima de $80Hz$, garantindo precisão e tempo de resposta adequados para a aplicação em questão.

Adicionalmente aos requisitos relevantes ao tema abordado nesse trabalho, a placa de circuito impresso desenvolvida inclui um conector dedicado para o controle de um servomotor capaz de abrir e fechar a garra do robô e uma entrada e uma saída digital para possíveis expansões futuras. A alimentação da PCB é fornecida por uma fonte externa de 5V, com a conexão entre ambas realizada por meio dos cabos de passagem internos do robô, que ligam a base à flange da ferramenta de trabalho.

A Figura 15 apresenta a parte superior do projeto da PCB final enviado para fabricação. Podem ser vistos os componentes necessários para a leitura da célula de carga e os conectores utilizados para a alimentação, controle do motor e conexão do sensor.

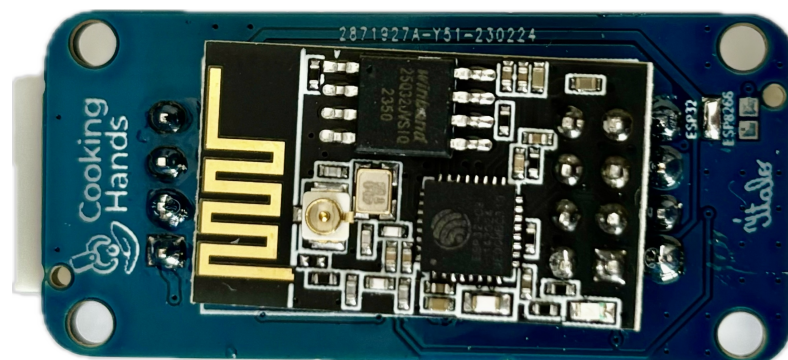
Figura 15 – PCB para leitura da célula de carga



Fonte: Autor (2024).

O microcontrolador selecionado foi o ESP32-C3, da Espressif, devido ao seu baixo custo, excelente conectividade Wi-Fi e ótimo desempenho em sistemas operacionais de tempo real (BASSAN, 2021). Para reduzir os custos de produção em pedidos de pequenas quantidades de placas, optou-se por utilizar o microcontrolador em uma placa de desenvolvimento disponível comercialmente, como mostrado na Figura 16.

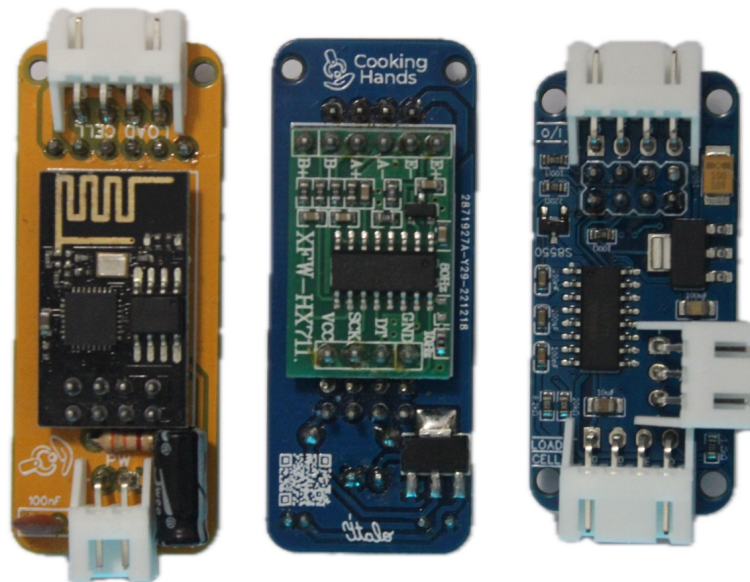
Figura 16 – Vista inferior da PCB com a placa de desenvolvimento



Fonte: Autor (2024).

Além dos requisitos funcionais, existiu para o desenvolvimento dessa placa de circuito impresso, a necessidade de manter as dimensões reduzidas para que fosse possível manter inalterado o aspecto visual da garra do braço robótico onde esse dispositivo foi instalado. Após o desenvolvimento e validação do primeiro protótipo, os sistemas necessários foram integrados em uma versão definitiva da placa, com a utilização de dispositivos de montagem superficial, cujo esquemático pode ser encontrado no Apêndice A. Na Figura 17, são apresentadas as PCBs produzidas com componentes *through-hole* à esquerda e ao centro, enquanto a placa com componentes de montagem superficial está à direita.

Figura 17 – Comparação das placas produzidas com componentes THT e SMD



Fonte: Autor (2024).

4.1.2.5 Firmware do microcontrolador

O programa do microcontrolador foi desenvolvido com o *framework* Arduino devido a sua popularidade e ampla compatibilidade com bibliotecas de comunicação e leitura de sensores (BANZI, 2011). Foi utilizado o sistema operacional de tempo real FreeRTOS para gerenciar as tarefas responsáveis por realizar a leitura da célula de carga, comunicação de rede e recebimento de comandos de forma concomitante (BARRY, 2016). O código completo desenvolvido pode ser encontrado no Apêndice B.

A comunicação Wi-Fi foi escolhida devido aos problemas de interferência encontrados ao enviar sinais de comunicação através dos cabos de passagem dentro do robô, que conectam sua base à garra. Durante os testes preliminares, a comunicação cabeada era possível apenas quando os *drivers* dos motores do braço robótico estavam desabilitados. Dessa forma, não seria possível utilizá-la durante o processo de servir o

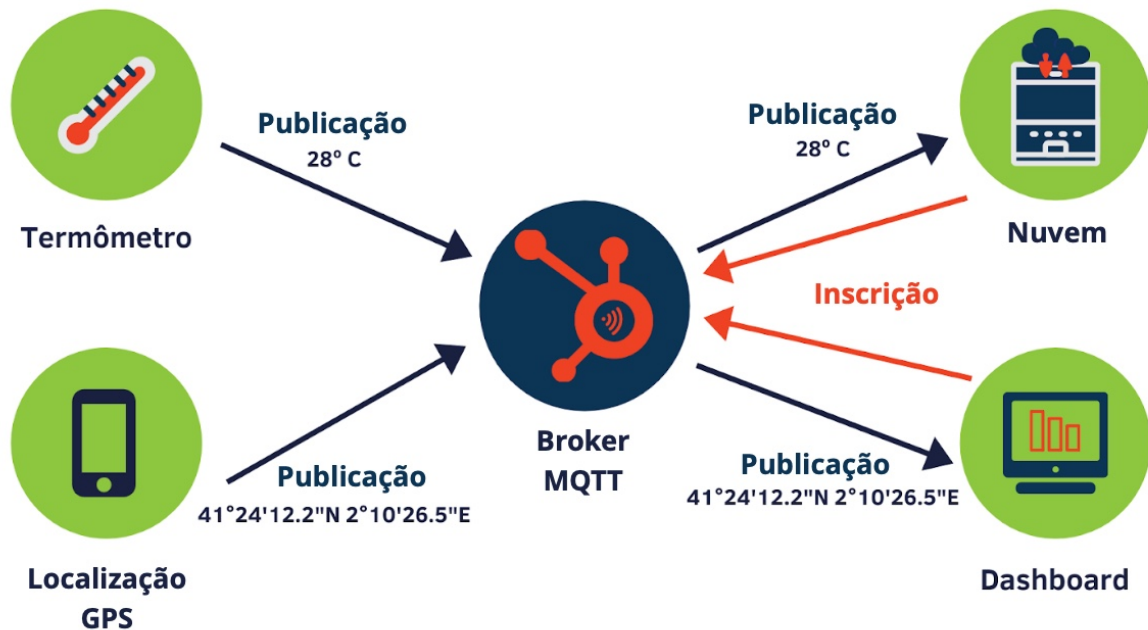
sorvete, quando o robô precisa realizar um movimento circular para distribuir o produto por todo o copo.

4.1.2.6 Propagação dos dados

Os dados de peso coletados pela PCB, responsável pela célula de carga, precisam ser transmitidos ao computador de comando, que é responsável por enviar os comandos e dados relevantes ao sistema de controle das válvulas da máquina de sorvete, ao controlador do braço robótico, e à base de dados que armazena o histórico dos sorvetes produzidos.

Para essa comunicação, foi utilizado o protocolo *Message Queuing Telemetry Transport* (MQTT)¹, escolhido por sua simplicidade de uso e alta velocidade na transmissão de dados. O protocolo é baseado em uma arquitetura composta por um dispositivo responsável por receber e distribuir todas as mensagens, chamado de *broker*, e dispositivos conectados a ele, chamados de clientes. Os clientes podem publicar mensagens em tópicos e, também, se inscrever em tópicos dos quais desejam receber as mensagens postadas por outros clientes (USMANI, 2021). A Figura 18 exemplifica os componentes de um sistema genérico de distribuição de dados utilizando o protocolo MQTT.

Figura 18 – Exemplo da arquitetura de um sistema de distribuição de dados utilizando MQTT



Fonte: Adaptado de Afonso (2020).

¹ www.mqtt.org

No sistema em questão, o *broker* é um programa que roda no computador de comando do quiosque e a PCB da garra, o programa de controle do robô e o programa responsável por armazenar os dados dos pedidos na base de dados são os clientes. A garra publica os dados de peso lidos pela célula de carga, e os demais clientes recebem esses dados para realizar suas respectivas funções.

4.1.2.7 Comunicação com o robô

Para o funcionamento desse sistema é essencial que o braço robótico saiba quando o peso desejado do sorvete foi atingido para que esse possa continuar com as demais etapas da produção de sorvete. Foi utilizada a ferramenta de código aberto KukavarProxy, que permite a leitura e alteração de variáveis globais no arquivo de configuração do controlador do braço robótico a partir de qualquer dispositivo conectado na rede local (FAGO; ROSA, 2019). Esse método foi escolhido principalmente por não exigir a compra de nenhum pacote de comunicação adicional desenvolvido pela fabricante do robô.

O controlador do braço robótico utiliza o sistema Windows para executar os programas de controle de movimentação e a interface do usuário. O programa KukavarProxy pode ser instalado no sistema operacional para agir como um servidor de interface entre outros dispositivos conectados à mesma rede e as variáveis globais do robô. Isso é possível pois essas variáveis são armazenadas em um arquivo de texto acessível pelo sistema de gerenciamento de arquivos do Windows.

4.2 DESENVOLVIMENTO DE SOFTWARE

4.2.1 Base de dados

As informações relevantes para o desenvolvimento deste trabalho, referentes a cada pedido, foram armazenadas em uma base de dados, juntamente com a foto do sorvete correspondente. A seguir, são listados os dados agregados pelo sistema:

- Data e hora do pedido;
- Tamanho do copo;
- Sabor do sorvete;
- Tempo de abertura da válvula da máquina de sorvete;
- Peso final do sorvete servido;
- Foto do sorvete.

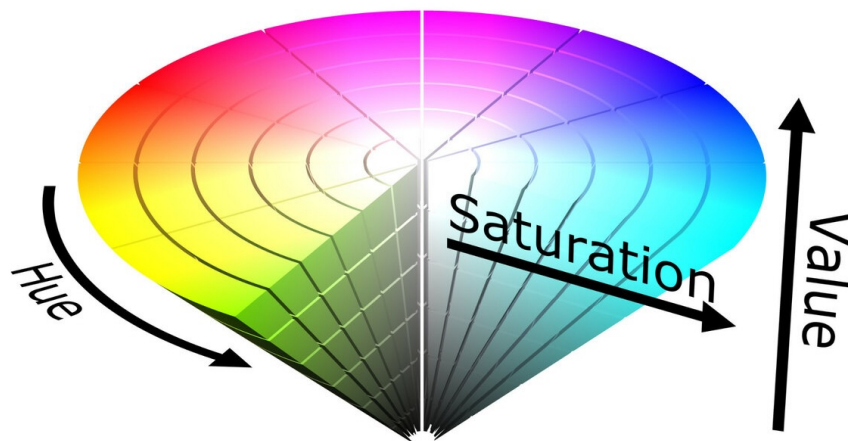
4.2.2 Visão computacional tradicional

A primeira abordagem utilizada para detecção do sorvete na imagem foi baseada em técnicas tradicionais de visão computacional, empregando ferramentas

de segmentação por análise de limites de cor, transformações morfológicas, filtros de realce e nitidez, assim como estratégias de equalização de histograma e correção de cor.

Devido à cor branca do objeto de interesse e à baixa qualidade da câmera responsável pela captura das imagens, foi escolhido o espaço de cores HSV, ilustrado na Figura 19, devido a sua representação dos *pixels* em valores independentes para a tonalidade (*hue*), saturação (*saturation*) e intensidade (*value*). Essa característica, faz com que seja possível identificar objetos de uma cor específica com mais facilidade em imagens com diferentes condições de iluminação (KANG et al., 2021).

Figura 19 – Representação de uma cor no espaço HSV



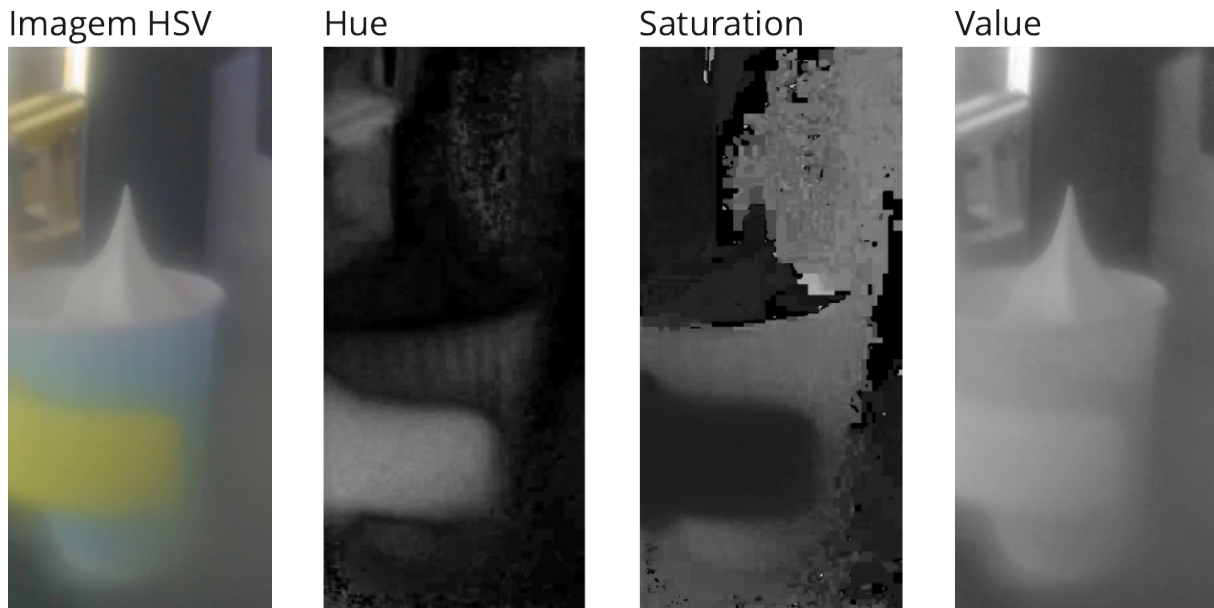
Fonte: Hyperskill (2024).

4.2.2.1 Limites de cor

A primeira etapa realizada para a detecção do sorvete na imagem, foi a criação de uma máscara explicitando as regiões da imagem onde a cor dos *pixels* se encontra dentro do intervalo definido para a cor do sorvete. Esse processo precisou ser realizado através de tentativa e erro, testando os valores definidos para os limites em imagens de amostra da base de dados.

Na Figura 20, pode ser visto que o sorvete pode ser encontrado nas regiões da imagem onde os valores dos canais de *hue* e *saturation* são próximos de zero e o de *value* é elevado. Já a garra, é evidenciada no canal de saturação por uma região escura e, principalmente, no canal de tonalidade pela região mais clara. Os objetos de interesse podem ser encontrados nas regiões onde os todos os canais de cores dos *pixels* se encontram dentro de um intervalo representativo das características descritas.

Figura 20 – Representação em escala de cinza dos canais de tonalidade, saturação e intensidade



Fonte: Autor (2024).

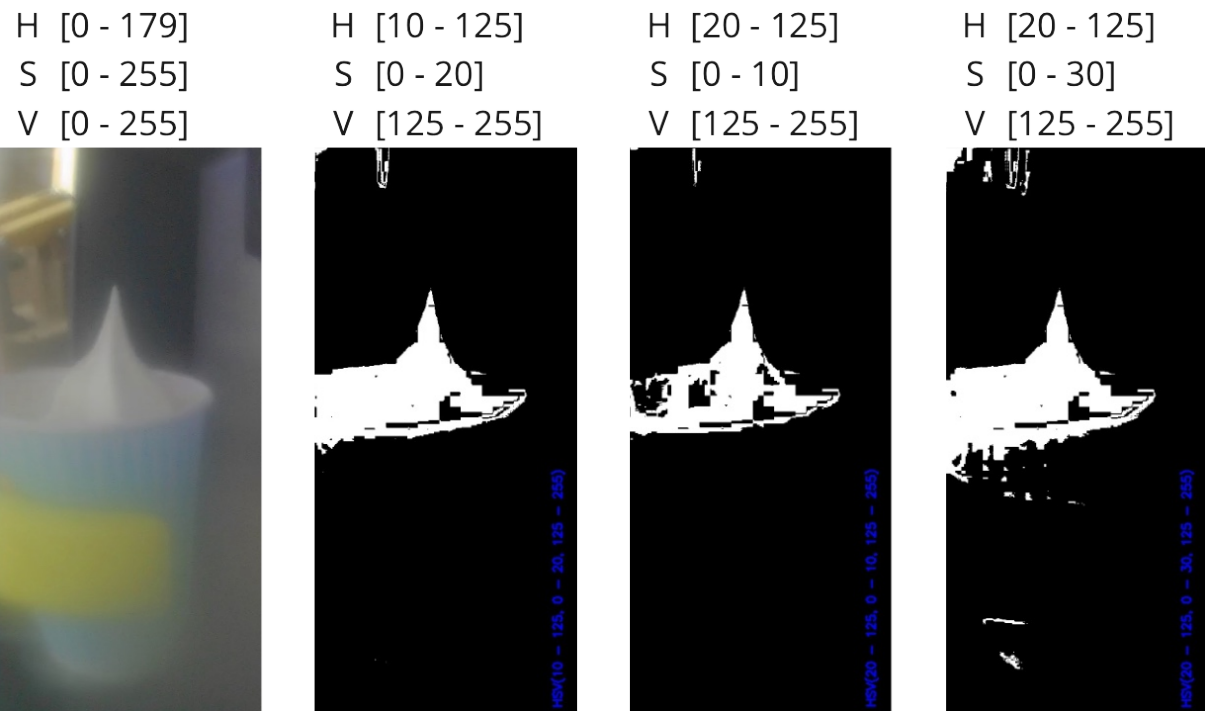
A seleção de limites muito amplos facilita a detecção do sorvete, contudo, acaba incluindo regiões irrelevantes da imagem na máscara. Portanto, é importante aplicar técnicas adicionais de tratamento de imagem e pós-processamento da máscara gerada de forma a minimizar a amplitude dos valores considerados. Na Figura 21, são apresentadas as máscaras geradas com a detecção de cores para alguns dos limites considerados.

4.2.2.2 Transformações morfológicas

A máscara obtida pela detecção de cores pode apresentar buracos e bordas irregulares que não representam fielmente o objeto de interesse. Para mitigar esse problema, foi utilizada a transformação morfológica de fechamento, que consiste em uma operação de dilatação seguida de uma operação de erosão. Variando a quantidade de iterações e experimentando com *kernels* de formatos e tamanhos distintos foi possível estabelecer um procedimento para suavizar as bordas e preencher as lacunas da máscara.

A utilização de *kernels* de dimensões elevadas facilita a remoção de ruídos nas bordas e o preenchimento de lacunas, mas reduz os detalhes da máscara e remove regiões pontiagudas, o que é problemático para a detecção do contorno do sorvete. Os *kernels* de forma retangular têm ação uniforme em todas as direções, contudo, resultam em máscaras serrilhadas. Já os elementos em formato de cruz apresentam resultados mais suaves em máscaras de geometria sem características predominantes, porém são menos efetivos na redução de imperfeições.

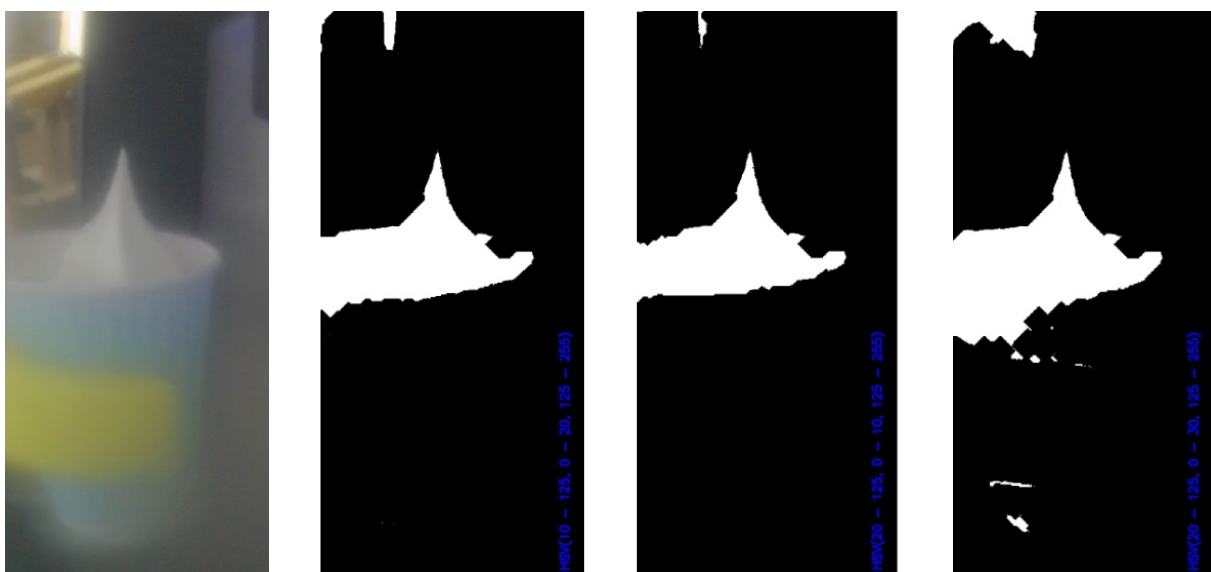
Figura 21 – Comparação entre máscaras com limites diferentes para a cor branca



Fonte: Autor (2024).

Na Figura 22 observa-se o resultado de 10 iterações da transformação de fechamento morfológico, utilizando um *kernel* em formato de cruz de 5 por 5 *pixels*, aplicado às máscaras mostradas na Figura 21.

Figura 22 – Resultado da transformação morfológica de fechamento nas máscaras



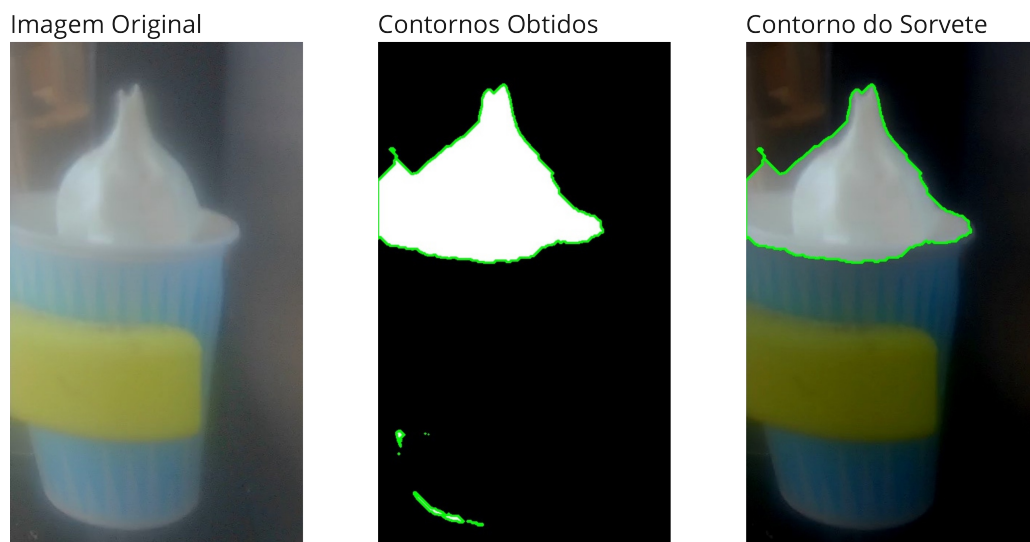
Fonte: Autor (2024).

4.2.2.3 Seleção do contorno

A partir da máscara obtida para imagem evidenciando as regiões que estão dentro do limite de cor definido, é possível obter um contorno para as regiões de interesse. Esse contorno é caracterizado pela curva fechada que une todos os pontos da fronteira entre duas regiões de cores distintas (RATH, 2021).

Como pode ser visto na Figura 23, foram obtidos diversos contornos para a seleção de cores referente ao sorvete e a garra. Em ambos os casos foi possível calcular a área de todos esses contornos e inferir que o objeto de interesse é representado pelo de maior área.

Figura 23 – Contornos obtidos para o limite de cores do sorvete



Fonte: Autor (2024).

4.2.2.4 Equalização de histograma e realce de contraste

A segmentação de objetos por cor apresenta, por natureza, um péssimo resultado quando aplicada em imagens com características de iluminação diferentes daquelas utilizadas para definir os limites de cor. A fim de mitigar os efeitos da variação de luminosidade, pode ser aplicada a técnica de equalização de histograma aliada ao realce de contraste.

Na Figura 24 pode ser visto o antes e depois de duas imagens com características distintas de iluminação de cena. Foi utilizado o método CLAHE (*Contrast Limited Adaptive Histogram Equalization*), nativo da biblioteca OpenCV, devido a sua capacidade de explicitar detalhes da imagem sem introduzir ruído excessivo (ROSEBROCK, 2021a). Apesar do método ter sido desenvolvido para imagens em escala de cinza, pode ser aplicado individualmente para cada um dos canais de cor de uma imagem colorida.

Figura 24 – Exemplos da equalização de histograma e aprimoramento de contraste



Fonte: Autor (2024).

4.2.3 Modelo de segmentação contextual

O primeiro modelo de segmentação contextual escolhido foi o *Segment Anything* (SAM), desenvolvido pela empresa Meta e disponibilizado gratuitamente para uso público. Esse modelo identifica objetos e regiões individuais em uma imagem a partir da entrada de pontos ou regiões de interesse onde o sujeito se encontra. Além disso, é possível indicar pontos conhecidamente fora da região de interesse para melhorar os resultados em situações ambíguas (KIRILLOV et al., 2023).

Figura 25 – Objetos e regiões detectadas pelo modelo *Segment Anything*



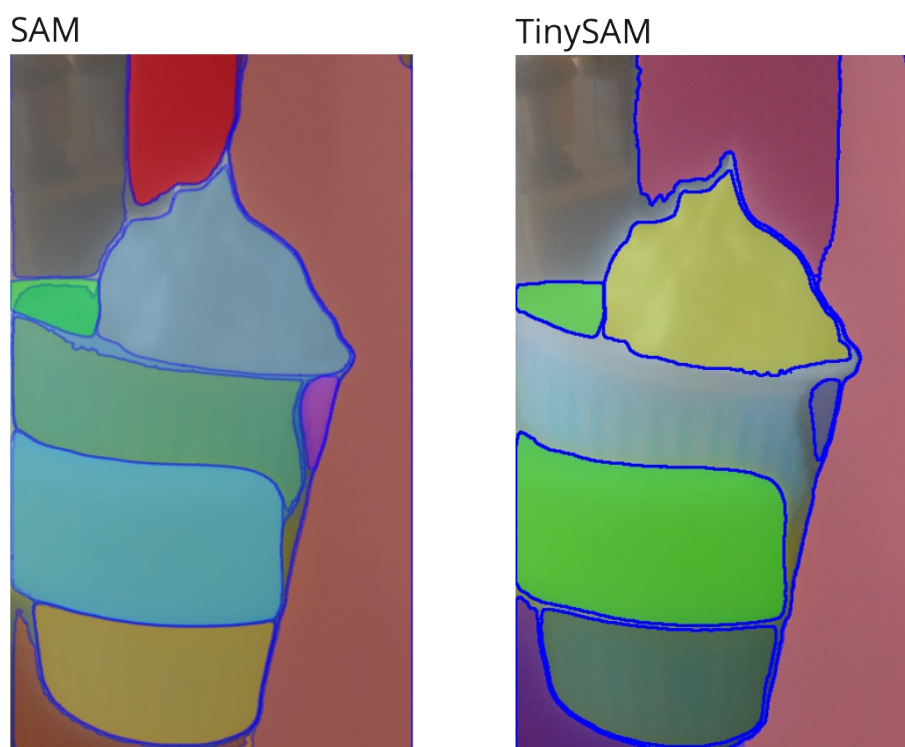
Fonte: Autor (2024).

Para verificar a viabilidade do SAM para a aplicação em questão, foi utilizado o *website* de demonstração da ferramenta. Na Figura 25, podem ser vistos todos os objetos individuais detectados pelo modelo em uma das imagens da base de dados, o sorvete e a garra foram precisamente identificados apesar da baixa qualidade da foto.

Embora o modelo apresente resultados excelentes, o tempo de execução em um computador pessoal foi extremamente alto, na ordem de minutos para uma única imagem. Assim, a execução no computador de controle dos quiosques da Roboteria seria inviável, pois comprometeria o desempenho dos programas essenciais para a operação e não permitiria ajustes contínuos na dosagem do sorvete.

A fim de mitigar o problema do custo computacional foi utilizado o modelo TinySAM, desenvolvido para apresentar resultados semelhantes ao SAM porém com tempo de execução extremamente reduzido (SHU et al., 2023). Nos testes realizados, a ferramenta foi capaz de identificar os contornos correspondentes ao sorvete e a garra em apenas dois segundos de execução de forma consistente. Na Figura 26, pode ser observado que, apesar de apresentar menos detalhes que o modelo original, o TinySAM consegue detectar os objetos de interesse com boa exatidão e precisão.

Figura 26 – Comparação entre os resultados do SAM e TinySAM

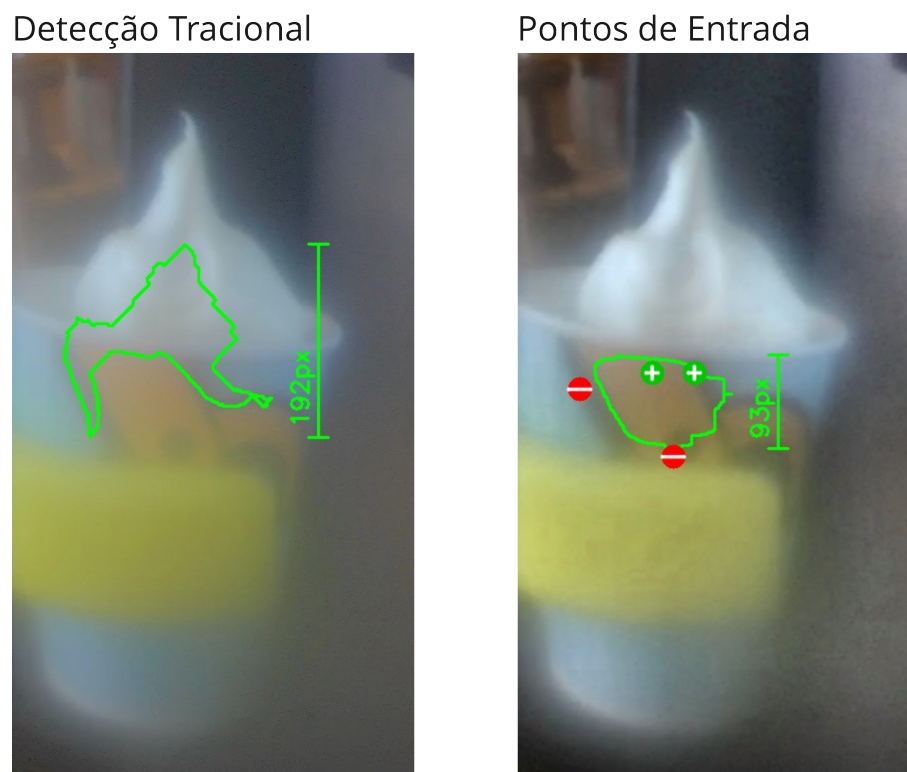


Fonte: Autor (2024).

4.2.3.1 Seleção dos pontos de entrada

A primeira estratégia utilizada para a seleção dos pontos de entrada para o modelo de segmentação foi utilizar o contorno obtido pela visão computacional tradicional para calcular um ponto médio onde o sorvete e a garra deveriam estar presentes. Contudo, boa parte dos resultados obtidos pela abordagem tradicional erraram completamente a posição do sorvete, fazendo com que os pontos de entrada não correspondessem ao objeto de interesse. Esse modo de falha é exemplificado na Figura 27.

Figura 27 – Exemplo de falha na seleção automática de pontos de entrada



Fonte: Autor (2024).

Essa estratégia foi preferida de início por ter a possibilidade de levar em conta diferenças no posicionamento da câmera e possíveis desalinhamentos dos mecanismos durante a operação, limpeza e manutenção dos quiosques. Contudo, devido aos resultados insatisfatórios obtidos e a pequena variação de posição observada durante o período em que as imagens foram coletadas foi considerado satisfatório a configuração manual de pontos de entrada para cada um dos quiosques.

Na Figura 28, pode ser observado que foram escolhidos pontos positivos, que representam a região onde espera-se encontrar o objeto de interesse e pontos negativos, que representam regiões onde não deve ser encontrado. A adição dos pontos negativos aprimorou a performance do modelo nos casos mais desafiadores,

onde as condições de iluminação dificultavam a separação entre o sorvete e o interior do copo ou quando o logo amarelo presente no copo se encontrava na vizinhança da garra.

Figura 28 – Pontos de entrada fixos para o modelo de segmentação contextual



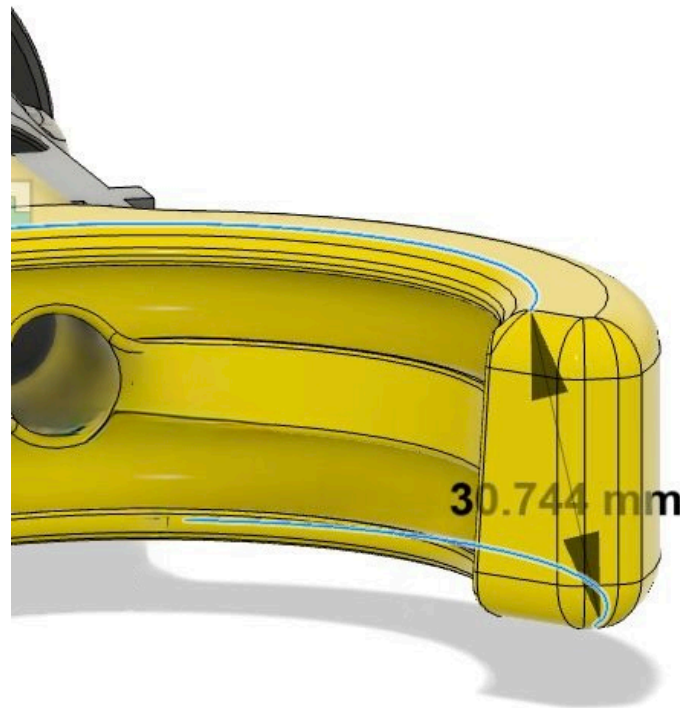
Fonte: Autor (2024).

4.2.4 Detecção da garra

Com o objetivo de obter uma medida de referência que permita a conversão do tamanho do sorvete em *pixels* para milímetros, foram utilizadas as mesmas estratégias de detecção do sorvete para encontrar o contorno da garra do robô na imagem. Esse objeto foi escolhido por se manter aproximadamente no mesmo plano do sorvete independentemente do posicionamento da câmera e possuir uma cor facilmente identificada e pouco prevalente em outras regiões da imagem.

A altura de referência para a garra foi obtida através do *software* de modelagem 3D. Como mostrado na Figura 29, a dimensão considerada foi a distância entre a borda superior esquerda da seção transversal da garra e a sua borda inferior direita.

Figura 29 – Dimensões da garra do robô



Fonte: Autor (2024).

Para obter um valor mais relevante para a altura da garra, foi calculada a média da distância entre a borda inferior e superior ao longo de todo o contorno obtido. Foram retirados os valores extremos do conjunto de dados contendo as distâncias para mitigar o efeito das imperfeições no contorno obtido. O método utilizado para remoção dos dados extremos foi o intervalo entre quartis, onde são considerados apenas os valores entre o limite inferior e superior calculados conforme as Equações (3) e (4) (TAYLOR, 2024).

$$Limite_{Inferior} = Q_1 - 1,5IQR \quad (3)$$

$$Limite_{Superior} = Q_3 + 1,5IQR \quad (4)$$

onde,

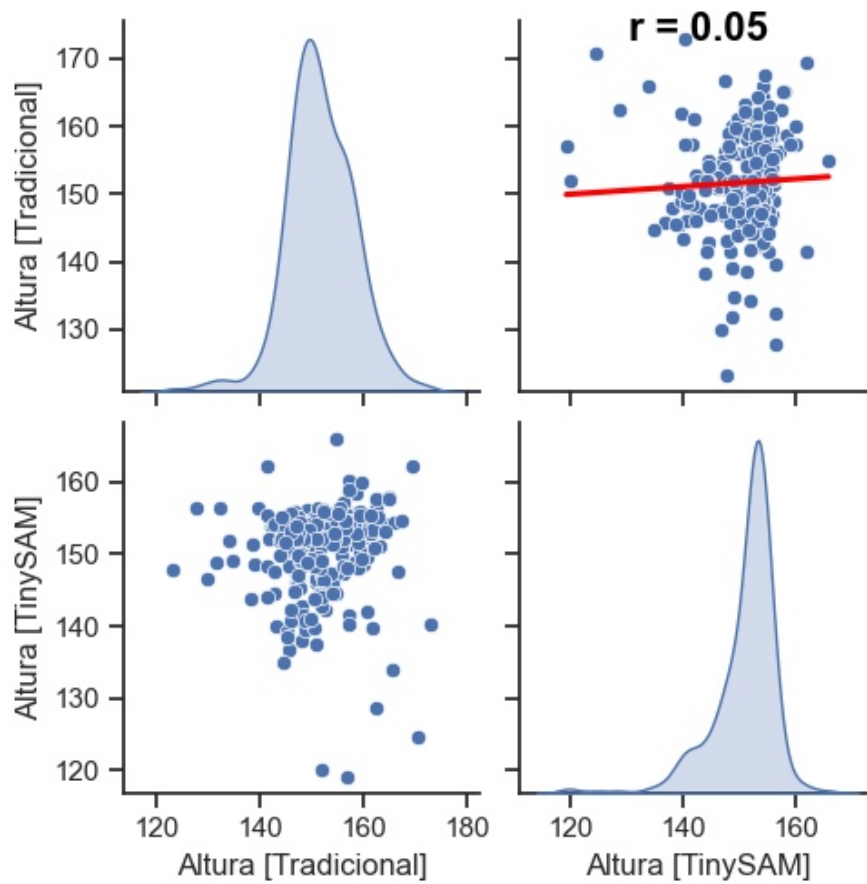
$$IQR = Q_3 - Q_1$$

sendo Q_1 o valor que separa os 25% menores valores do conjunto de dados e Q_3 o que separa os 25% maiores.

Na Figura 30, pode ser vista a distribuição e a correlação dos valores obtidos para a altura, em *pixels*, da garra utilizando a abordagem tradicional e por aprendizado de máquina. Os valores obtidos com o modelo de segmentação diferem significativamente daqueles gerados pela segmentação baseada em limites de cor.

A análise dos contornos obtidos mostra que o modelo de segmentação retornou contornos muito mais precisos e exatos para o sujeito de interesse, logo, apenas os valores calculados com base nele foram utilizados no desenvolvimento desse trabalho.

Figura 30 – Distribuição dos valores de altura da garra calculados utilizando a visão computacional tradicional e aprendizado de máquina



Fonte: Autor (2024).

5 RESULTADOS

5.1 ANÁLISE EXPLORATÓRIA DOS DADOS

O programa desenvolvido foi utilizado para analisar as imagens dos sorvetes de tamanho médio e sabor baunilha produzidos pela unidade da Roboteria em Joinville, no período de 26 de março a 19 de abril de 2024. Ao todo, foram inspecionadas 314 fotos de sorvetes distintos, que continham dados válidos de peso e tempo de dosagem registrados na base de dados com o histórico de pedidos da unidade.

Durante a operação autônoma do quiosque, sem a presença de um atendente humano, algumas condições adversas podem levar à produção de sorvetes inválidos. Quando o suprimento de massa da máquina de sorvete acaba, o robô produz um copo completamente vazio e, se o copo for amassado durante o processo de pega, o sorvete pode vazar ou escorrer pelas bordas. Nessas situações, exemplificadas na Figura 31, os contornos resultantes seriam inadequados e comprometeriam as avaliações. Por esse motivo, essas fotos foram removidas manualmente da base de dados antes da realização do estudo.

Figura 31 – Exemplos de imagens rejeitadas para análise

Sorvete Vazando



Copo Vazio

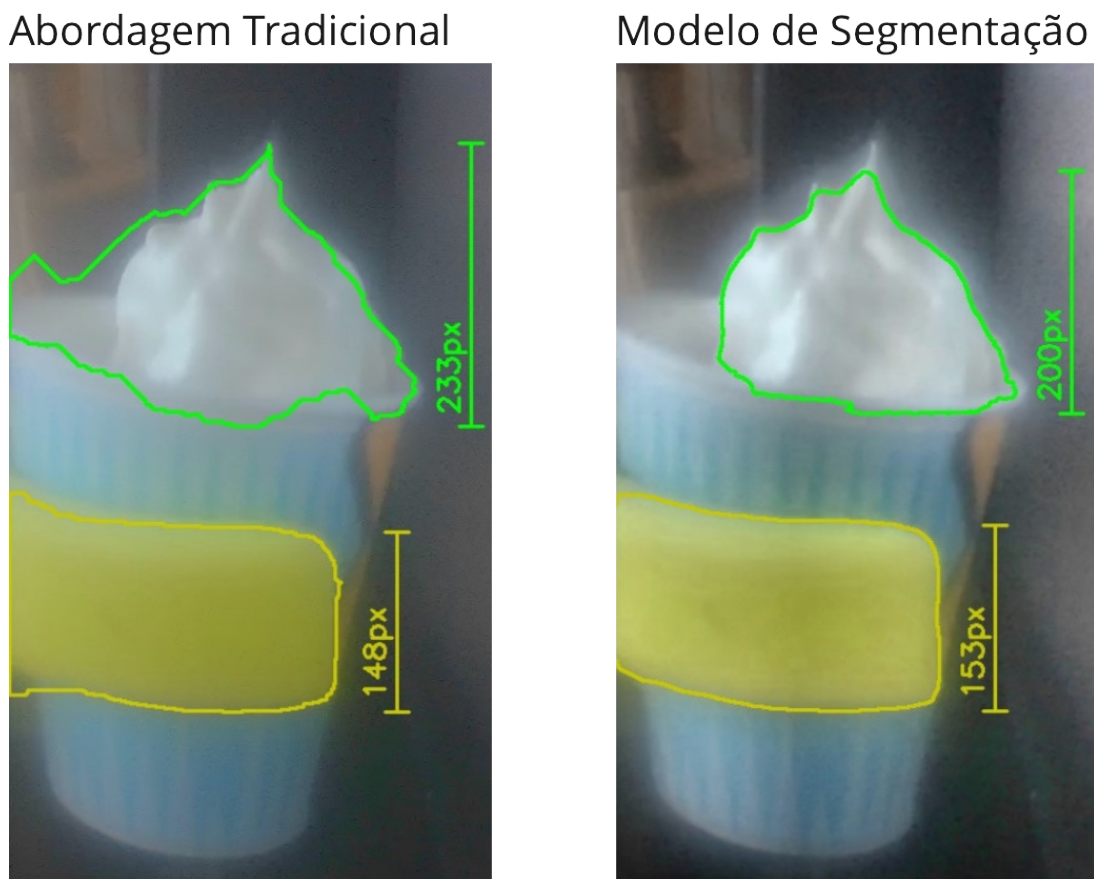


Fonte: Autor (2024).

5.1.1 Comparação entre abordagem tradicional e aprendizado de máquina

Para cada uma das imagens, o programa detectou o sorvete no copo e a garra do robô utilizando as técnicas de visão computacional tradicional abordadas no capítulo anterior e o modelo de segmentação contextual TinySAM. Foram geradas imagens, como a da Figura 32, comparando os contornos obtidos para os objetos de interesse de ambas as abordagens e foi criada uma planilha contendo os valores obtidos para a altura e área desses contornos.

Figura 32 – Exemplo de imagem de comparação entre a abordagem tradicional e aprendizado de máquina

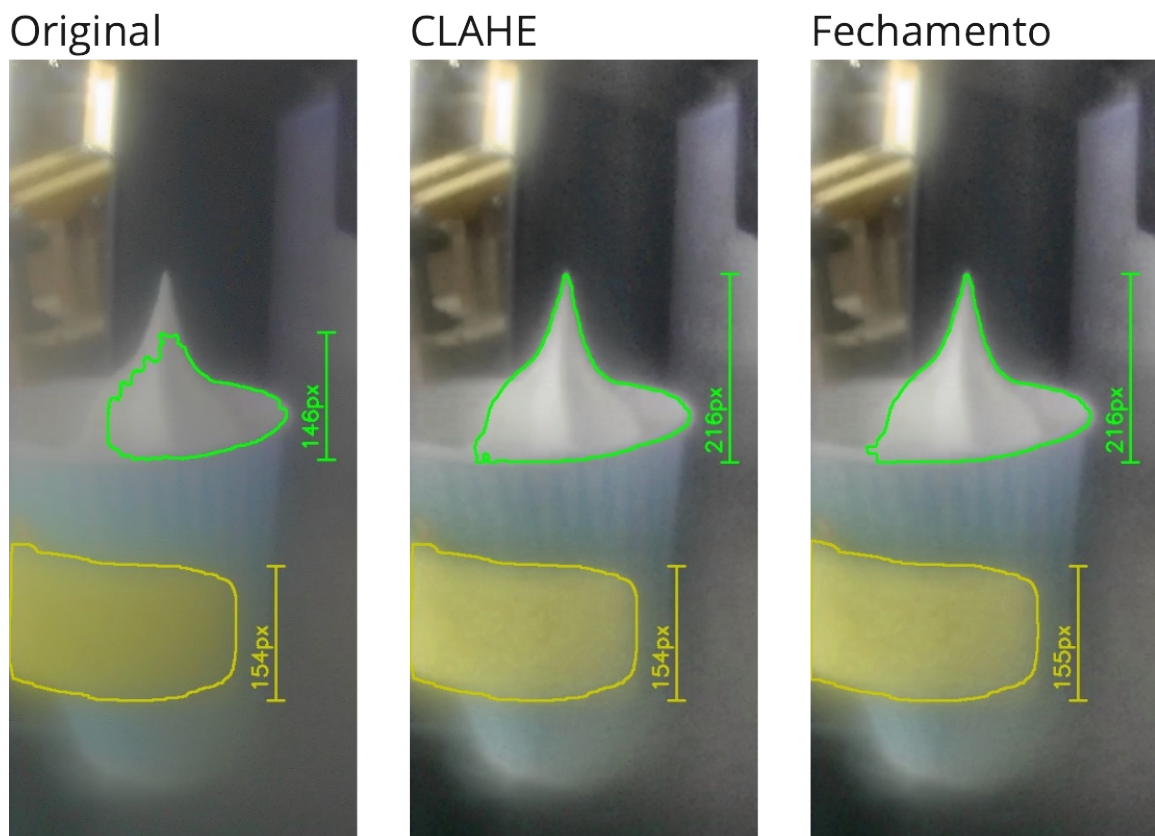


Fonte: Autor (2024).

Analisando visualmente as imagens de comparação obtidas, é possível constatar que a abordagem tradicional de visão computacional não apresentou bons resultados, incluindo áreas externas ao sorvete e artefatos de imagem ao contorno do objeto e, por vezes, falhando completamente em detectar o sujeito de interesse. A baixa qualidade das imagens fez com que não fosse possível definir um limite para os valores de cor que apresentasse bons resultados para todas as imagens, mesmo com a utilização de técnicas de pré-processamento de imagem e pós-processamento das máscaras obtidas.

Em contrapartida, o modelo de segmentação contextual baseado em aprendizado de máquina apresentou resultados consistentes para a detecção dos sorvetes e da garra do robô em todas as imagens analisadas. Como mostrado na Figura 33, os resultados desse método foram ainda mais precisos com a utilização do método de equalização de histograma e realce de contraste nas imagens de entrada e a aplicação da transformação morfológica de fechamento nas máscaras de saída.

Figura 33 – Exemplo de resultado obtido com técnicas de processamento



Fonte: Autor (2024).

5.1.2 Correlação entre *pixels* e milímetros

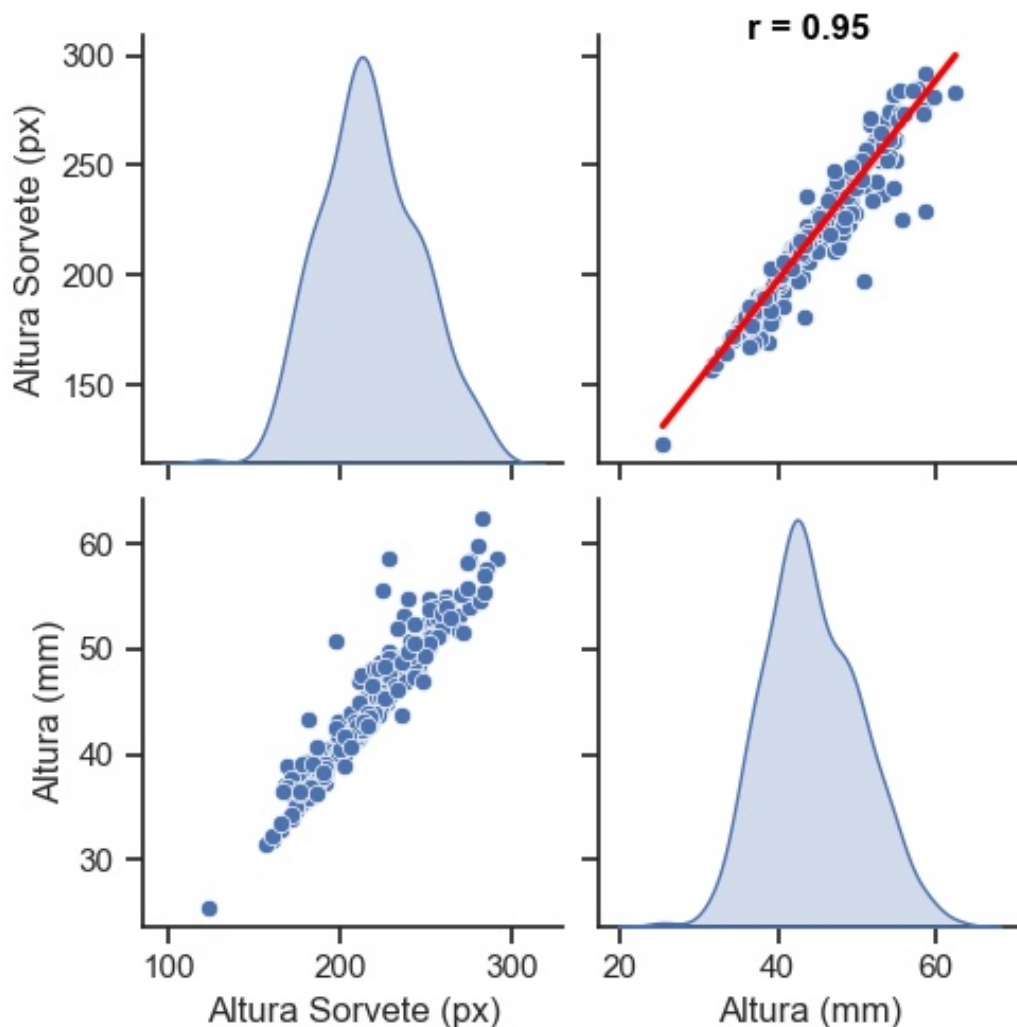
A fim de facilitar a definição dos parâmetros desejados para o sorvete em diferentes unidades da Roboteria, com diferentes características de instalação e equipamentos, é desejável que seja possível obter uma medida do produto proporcional ao seu tamanho real. Para que isso seja possível, é necessário obter uma relação para o tamanho físico de um *pixel* no plano contendo o sujeito de interesse.

Nesse trabalho, foi utilizada a medida da altura garra do robô como valor conhecido para o cálculo da correlação. O tamanho de um *pixel* foi então calculado dividindo a altura real da garra do robô pela altura em *pixels* obtida pelo programa em cada uma das imagens.

O valor obtido com esse método não representa exatamente a medida real do objeto, pois é baseado na qualidade da detecção do contorno da garra e na aproximação de que a garra e o sorvete se encontram suficientemente próximos um do outro para que os *pixels* representem uma mesma distância em ambas as vizinhanças. Para que fosse possível obter um resultado mais exato, seria necessário utilizar sorvetes de tamanho conhecido para calcular o tamanho físico dos *pixels* correspondente ao posicionamento específico da câmera em relação ao sujeito de interesse.

Contudo, essa abordagem foi considerada como um bom parâmetro de padronização para a aplicação em questão, uma vez que a garra se mantém a uma distância relativamente pequena e, principalmente, constante do sujeito de interesse. Na Figura 34, a correlação entre a altura do sorvete medida em *pixels* e o valor calculado em milímetros mostram uma relação bastante linear com poucos valores destoantes.

Figura 34 – Correlação entre altura do sorvete medida em *pixels* e o valor calculado em milímetros



Fonte: Autor (2024).

Um coeficiente de correlação igual a um representaria uma medida ideal da altura da garra e o posicionamento relativo constante entre essa e a câmera nas imagens analisadas. O valor $r = 0,95$ obtido permitiu inferir uma boa qualidade do contorno da garra fornecido pelo modelo de segmentação e baixa variabilidade no posicionamento dos componentes do sistema. Apesar de não garantir a exatidão dos resultados obtidos, a boa precisão alcançada permite que o método apresentado seja utilizado para padronizar as medidas calculadas para os sorvetes.

5.1.3 Correlação entre medidas e peso do sorvete

A altura do sorvete dificilmente poderia apresentar qualquer relação com o peso do produto, já que para um mesmo valor de altura são possíveis grandes variações no formato do sorvete. Contudo, foi considerada a possibilidade de que a área pudesse ser correlacionada com o peso do sorvete, já que essa é representativa do volume do produto.

Os resultados obtidos são compatíveis com a experiência prática de operação dos quiosques, a densidade dos sorvetes flutua significativamente durante o período de operação. Os principais fatores responsáveis por esse efeito são a temperatura e a quantidade de sorvete presente no cilindro da máquina de sorvete, assim como a quantidade de ar incorporada no produto pela bomba da máquina (GOFF; HARTEL, 2013).

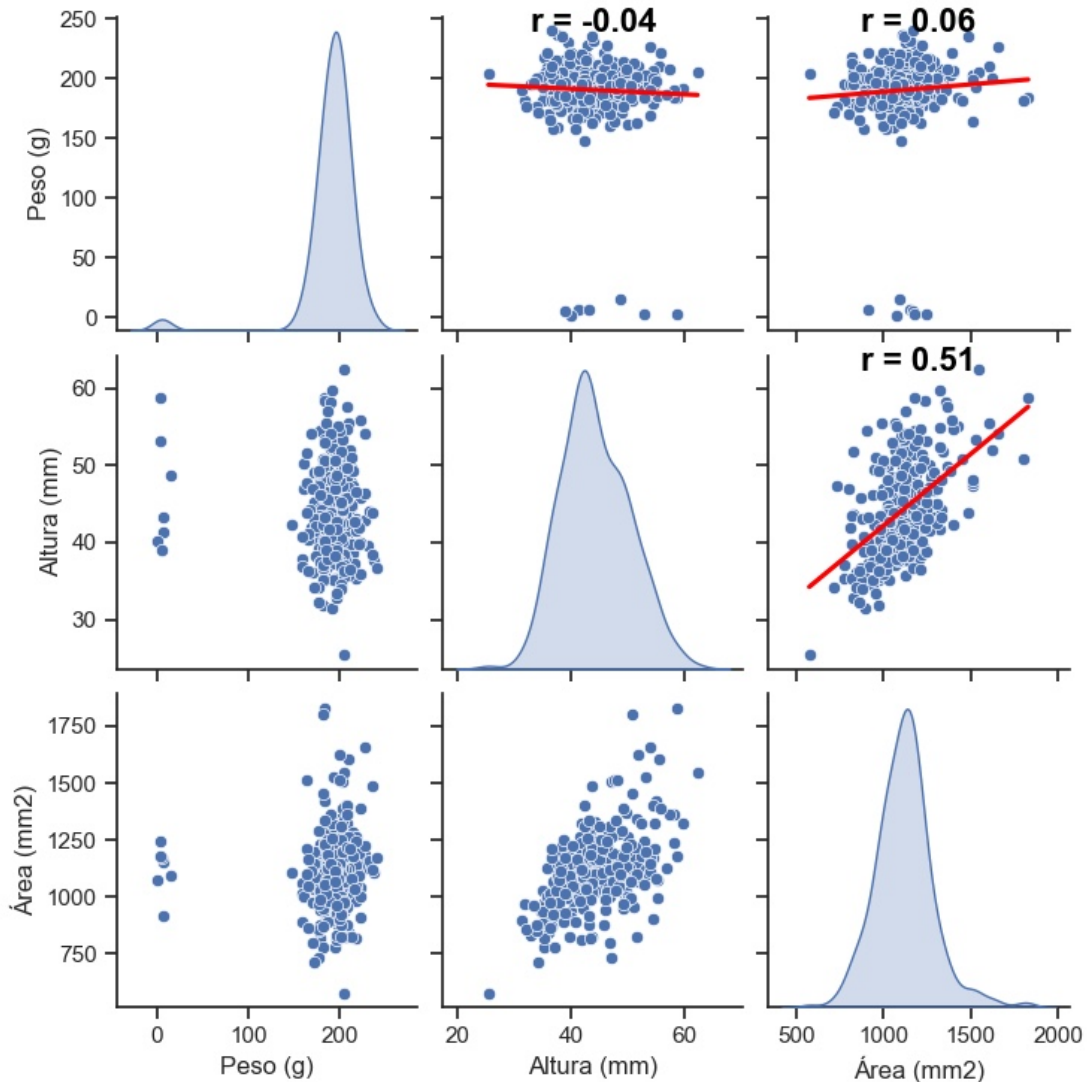
A máquina de sorvete possui ajustes para a definição dos valores desejados para os parâmetros mencionados anteriormente, contudo não possui nenhuma interface onde seja possível realizar a leitura do seu estado atual. Para que os parâmetros se estabilizem nos valores programados é necessário que a máquina fique um longo período sem produzir sorvetes, o que acarreta nas variações das características do produto observadas.

A Figura 35, apresenta a matriz de correlação entre os valores obtidos para altura e área do sorvete com o peso registrado na base de dados. Pode ser visto que nenhum desses valores apresenta uma forte relação com a massa, o que é evidenciado pelo baixo valor obtido para o coeficiente de correlação entre as variáveis. Dessa forma, não foi possível considerar a possibilidade de criar um modelo capaz de prever a altura ou área do sorvete com base no peso medido pela garra do robô com base nos dados disponíveis.

5.1.4 Correlação entre medidas e o tempo de dosagem do sorvete

O volume de sorvete servido para um determinado intervalo de abertura da válvula da máquina depende da velocidade com que ele é expelido. Essa velocidade, por sua vez, varia conforme a viscosidade do sorvete, que é influenciada principalmente

Figura 35 – Matriz de correlação entre altura, área e peso do sorvete



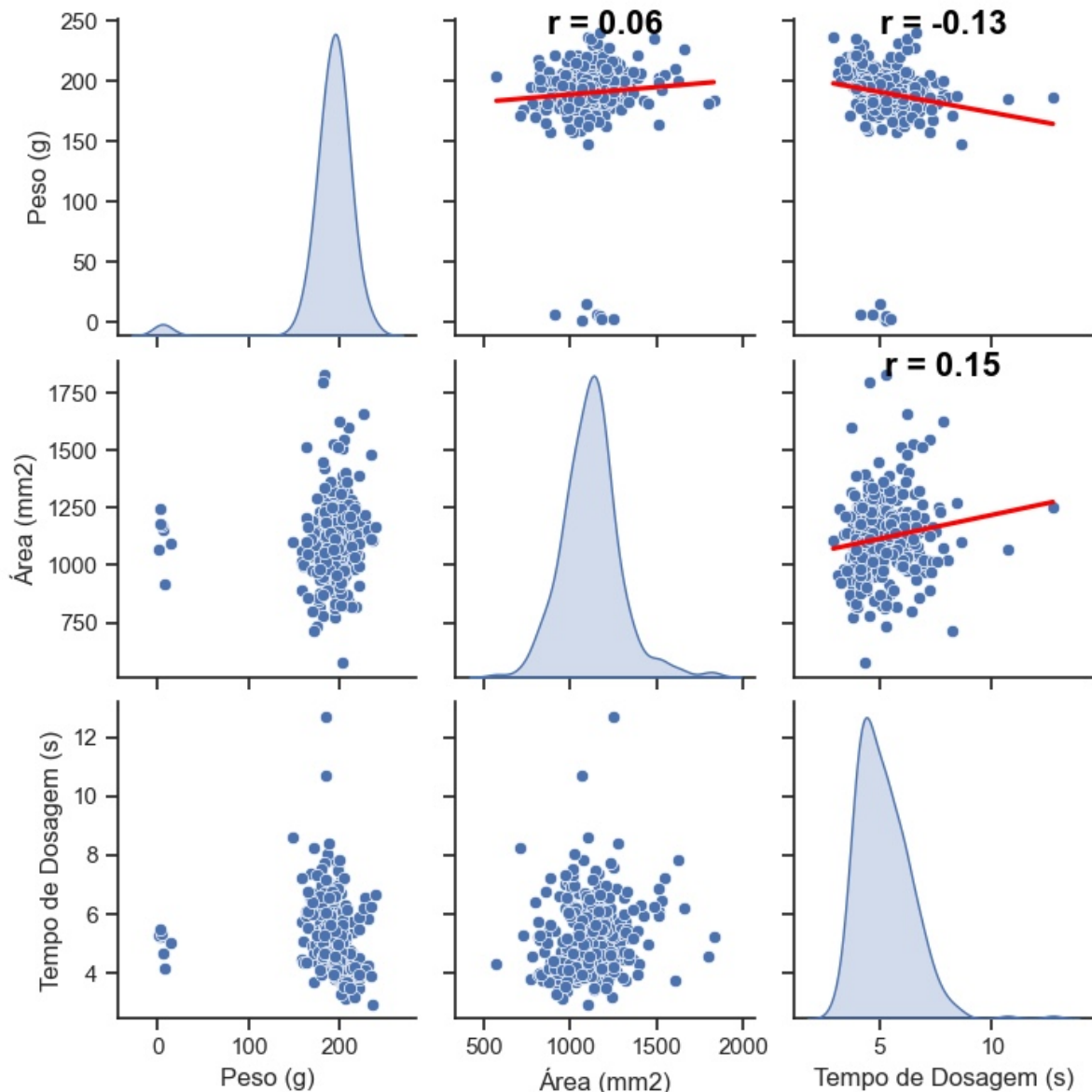
Fonte: Autor (2024).

pela temperatura do cilindro da máquina onde o sorvete é produzido.

A máquina possui um sistema de monitoramento da corrente consumida pelo motor responsável pela mistura da massa de sorvete com o ar no cilindro de resfriamento capaz de estimar a consistência do sorvete. Contudo, fatores externos como a temperatura e a frequência de serviço acarretam na flutuação da viscosidade final obtida ao longo da operação.

A Figura 36, mostra a relação entre o valor calculado para a área, o peso do sorvete e o tempo de abertura da válvula da máquina. Assim como observado nos resultados referentes ao peso, o tempo de dosagem não apresentou nenhuma relação direta com a quantidade de sorvete servido, seja em termos de volume ou massa. Esse fato demonstra a importância do sistema de dosagem por peso para garantir o serviço da quantidade correta do produto.

Figura 36 – Matriz de correlação entre área e peso do sorvete com o tempo de dosagem



Fonte: Autor (2024).

5.1.5 Sistema de dosagem por imagem

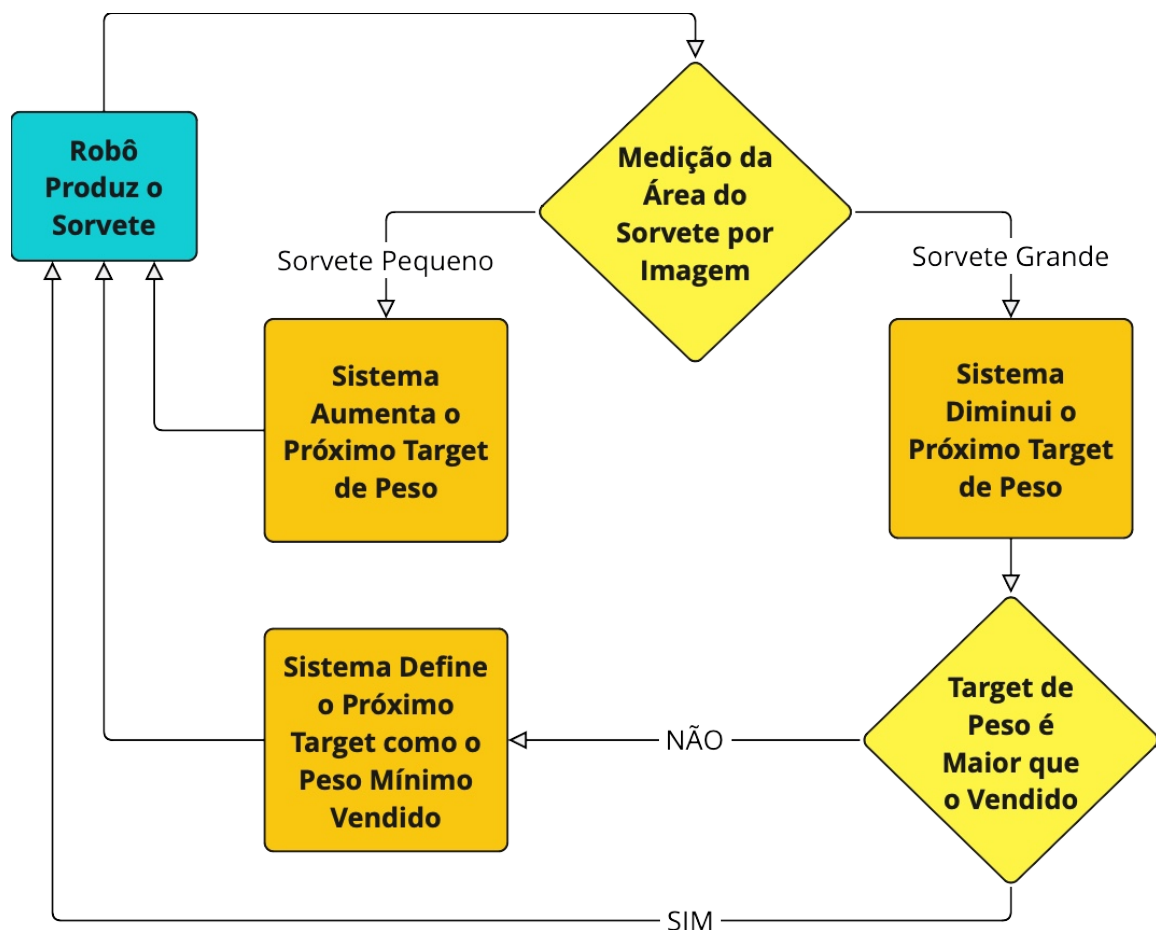
Como visto anteriormente nesse capítulo, não foi possível estabelecer uma relação entre os parâmetros de controle de dose do sorvete, tempo e peso, com a área do sorvete. Dessa forma, o sistema de medição por imagens poderá ser utilizado apenas como um fator de correção para o sistema de dosagem por peso. A dosagem por tempo é preterida em relação ao peso por não ser possível garantir uma quantidade mínima do produto para ser entregue ao cliente mensurável diretamente.

O Artigo 6º Inciso III do Código de Proteção e Defesa do Consumidor (BRASIL, 1990), exige que os todos os produtos vendidos tenham uma especificação clara de

quantidade, que deve ser assegurada a todos os consumidores. No caso de produtos que não preenchem completamente os recipientes onde são vendidos, como é o caso dos sorvetes de tipo *sundae*, é usual e justo que essa quantidade seja expressada em relação ao peso líquido do produto e não ao volume.

Utilizando a dosagem por peso, é possível garantir que o cliente receberá ao menos a quantidade de sorvete pela qual pagou, independentemente do funcionamento correto do sistema de visão ou de qualquer variação nos parâmetros de operação da máquina. O sistema, ilustrado na Figura 37, age como um auxílio na melhoria do aspecto de quantidade percebido visualmente pelos clientes, fornecendo mais massa de sorvete do que a comprada quando a densidade do produto estiver reduzida.

Figura 37 – Sistema de dosagem de sorvete por imagem



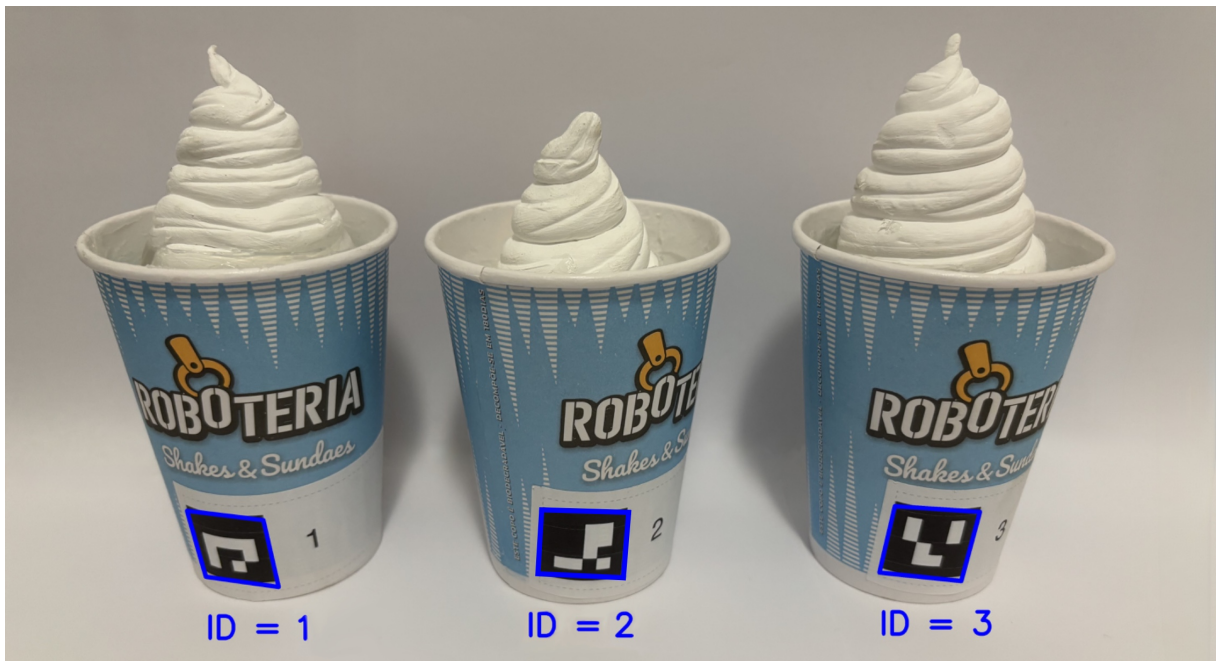
Fonte: Autor (2024).

5.2 CALIBRAÇÃO

Com o objetivo de avaliar a performance do sistema de medição dos sorvetes em relação a medida real dos mesmos, foi realizado um experimento com dez corpos

de prova. O produto real não poderia ser utilizado para medições físicas por ter uma consistência mole e derreter rapidamente. Portanto, foram produzidas artesanalmente réplicas de sorvete, a partir de porcelana fria, com um formato e coloração similares a um sorvete de baunilha mostradas na Figura 38.

Figura 38 – Copos com as réplicas de sorvetes



Fonte: Autor (2024).

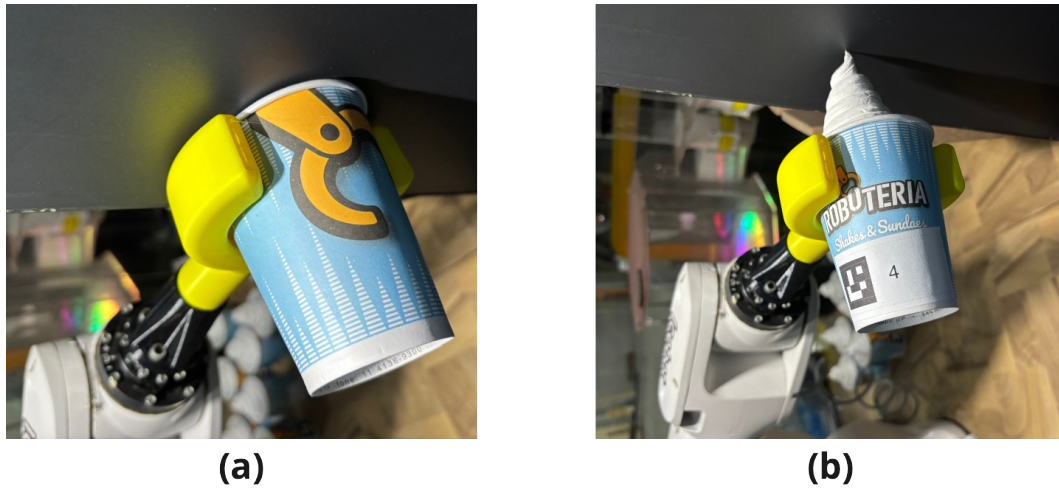
5.2.1 Medição dos corpos de prova

Para cada um dos corpos de prova, um copo foi colocado na garra do braço robótico. O robô então foi programado para se deslocar linearmente para cima até uma posição específica, onde o copo seria prensado para baixo na garra por um anteparo de posição conhecida e paralelo a borda da garra, como pode ser visto na Figura 39(a). Após esse processo, o copo fica firmemente posicionado com sua borda rente a garra.

Em seguida, o robô se afasta 200mm do anteparo para permitir que a réplica seja inserida e colada firmemente dentro do copo e o robô é movimentado manualmente até que a ponta do sorvete encoste no anteparo, como mostrado na Figura 39(b). Dessa forma, a altura da porção do corpo de prova que se encontra para fora do copo pode ser calculada subtraindo a distância total do movimento manual dos 200mm de afastamento da borda do copo em relação ao anteparo onde essa foi pressionada.

A Figura 39 mostra o braço robótico prensando o copo na garra e encostando a ponta da réplica do sorvete no anteparo para que sua posição possa ser utilizada na determinação da altura do corpo de prova.

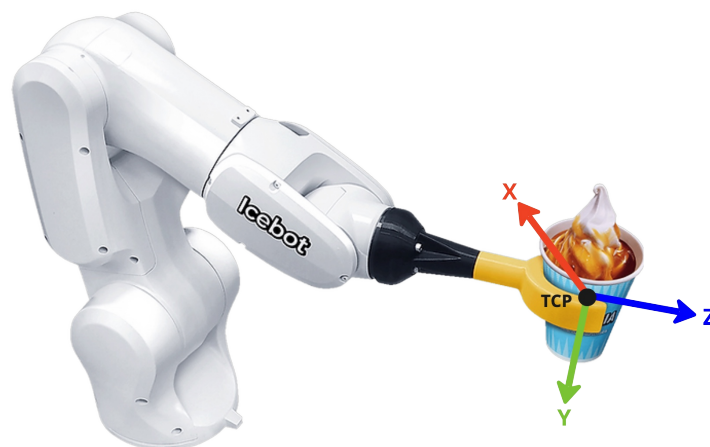
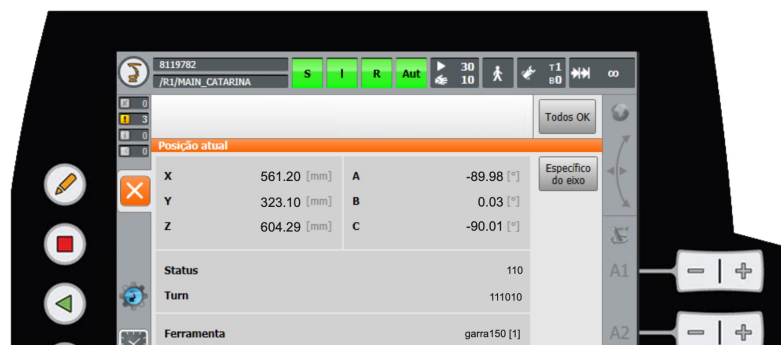
Figura 39 – Processo de medição dos corpos de prova com o braço robótico



Fonte: Autor (2024).

A Figura 40 apresenta a visualização da posição cartesiana do ponto central da ferramenta (TCP) no controle do braço robótico e a orientação dos eixos cartesianos. O TCP da garra foi configurado como o centro da secção circular que suporta o copo.

Figura 40 – Posição do ponto central da ferramenta do robô



Fonte: Autor (2024).

O valor obtido para a altura em milímetros das réplicas, identificadas por um número de um a dez, pode ser visto na Tabela 1.

Tabela 1 – Altura dos corpos de prova

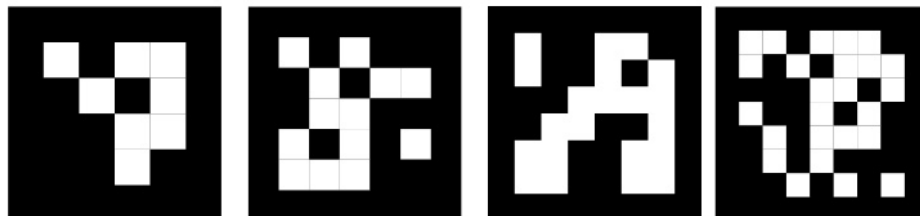
Número	Altura (mm)
1	43
2	31
3	50
4	61
5	48
6	38
7	52
8	44
9	47
10	56

Fonte: Elaborado pelo autor (2024).

5.2.2 Captura das imagens

Logo após o procedimento descrito na seção anterior, o robô foi posicionado na mesma posição utilizada para a captura da imagem dos sorvetes reais produzidos durante a operação. Foi, então, armazenada uma foto de cada um dos corpos de prova obtida com uma câmera posicionada na mesma posição e de mesmo modelo utilizado nos quiosques. Foi colado, em cada um dos copos, um adesivo marcador contendo o número de identificação do corpo de prova. Com o objetivo de facilitar um possível processo de calibração automatizado no futuro, foi utilizado um marcador do tipo ArUco, exemplificado na Figura 41, que pode ser facilmente identificado nas imagens pela biblioteca OpenCV (NAYAK, 2020).

Figura 41 – Exemplos de marcadores ArUco



Fonte: Siki e Takacs (2021)

5.2.3 Medição por imagem dos corpos de prova

O mesmo programa desenvolvido para a medição dos sorvetes salvos na base de dados de pedidos foi utilizado para calcular a altura em *pixels* e milímetros dos corpos de prova a partir das imagens capturadas. Como pode ser visto na Tabela 2, as medidas realizadas pelo programa, em sua maior parte, diferem muito dos valores de referência e o erro percentual é bastante inconsistente.

Tabela 2 – Comparação entre a altura de referência e a medida pelo programa

Número	Referência (mm)	Medida (mm)	Erro (%)
1	43	51,6	20,0
2	31	45,4	46,5
3	50	52,6	5,2
4	61	60,3	-1,2
5	48	53,6	11,7
6	38	46,0	21,1
7	52	58,8	13,1
8	44	53,3	21,1
9	47	53,3	13,4
10	56	59,8	6,8

Fonte: Elaborado pelo autor (2024).

A Figura 42 exemplifica os resultados obtidos para a medição por imagem dos corpos de prova. A medida obtida para o corpo de prova 2 apresentou o maior erro entre as amostras, o que foi causado pela falha mecânica da garra do robô que acarretou em uma diferença notável no posicionamento relativo do sorvete em relação a câmera. Por esse motivo, esse corpo de prova não foi considerado nas análises subsequentes.

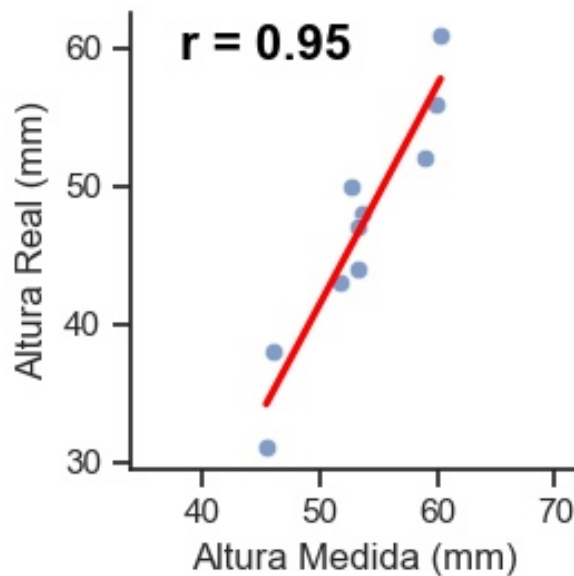
Figura 42 – Exemplos de resultados da medição das réplicas de sorvete



Fonte: Autor (2024).

Conforme pode ser observado na Figura 43, a relação entre as medidas obtidas pelo programa de medição dos sorvetes por imagem e a altura de referência dos corpos de prova analisados apresenta uma boa característica linear, com um coeficiente de correlação de Pearson igual a 0,95.

Figura 43 – Correlação entre a referência e as medidas dos corpos de prova



Fonte: Autor (2024).

Dessa forma, foi possível calcular um fator de correção para as medidas realizadas pelo programa a partir da inclinação da reta de regressão linear obtida. Os resultados obtidos para a altura do soverte a partir do modelo de segmentação contextual foram multiplicados pelo fator de correção. Na Tabela 3, pode ser observada uma redução significativa nos erros percentuais de medição após a aplicação do fator de correção.

Tabela 3 – Comparação entre a altura de referência das amostras e a medida corrigida

Número	Referência (mm)	Medida (mm)	Erro (%)
1	43	43,5	1,2
2	31	38,3	23,6
3	50	44,4	-11,3
4	61	55,6	-8,9
5	48	44,9	-6,6
6	38	39,2	3,0
7	52	49,1	-5,7
8	44	45,3	2,8
9	47	45,1	-4,1
10	56	50,2	-10,3

Fonte: Elaborado pelo autor (2024).

Como mostrado na Figura 44, apenas uma pequena porção da imagem capturada pela câmera é utilizada nas medições. A lente com grande campo de visão faz com que os objetos mais afastados do centro sofram distorções nas suas dimensões. Esse fator, aliado com a menor quantidade de *pixels* representando os objetos de interesse, é um dos responsáveis pelo erro de medição elevado encontrado.

Figura 44 – Comparação entre a imagem completa capturada e a região de interesse analisada

Imagem Completa



ROI



Fonte: Autor (2024).

6 CONCLUSÕES

Durante o desenvolvimento do algoritmo de detecção do sorvete utilizando a técnica de segmentação por cores, os principais desafios encontrados foram a baixa qualidade das imagens capturadas pela câmera do quiosque e as variações nas condições de iluminação. Apesar de testes extensivos com técnicas de pré-processamento de imagem, não foi possível encontrar um método com bons resultados para a detecção do sujeito de interesse.

Por esse motivo, optou-se por utilizar o modelo de segmentação contextual de imagem, *Segment Anything*, baseado em *deep learning*, que apresentou excelentes resultados nos testes preliminares. Contudo, esse modelo possui um custo computacional muito elevado, com tempo de execução incompatível com o tempo de ciclo da aplicação. A fim de solucionar esse problema, foi utilizado o modelo *TinySAM*, desenvolvido especialmente para situações onde a capacidade e tempo de processamento é limitada.

Para o desenvolvimento do sistema de leitura e dosagem de peso o principal desafio foi a interferência induzida nos cabos de comunicação que ligam a base do robô a sua flange, onde a garra com o sensor é instalada. Isso fez com que não fosse possível implementar um sistema de comunicação cabeado entre o controlador do braço robótico e a PCB responsável pela leitura do sensor. Dessa forma, foi necessário enviar os dados de peso via Wi-Fi para o computador de comando do quiosque que, por sua vez, transfere as informações para o robô.

Ao final desse trabalho foi possível desenvolver um sistema de visão computacional capaz de identificar o sorvete nas imagens e, a partir do contorno obtido, calcular a sua área e sua altura. Em conjunto com o sistema de dosagem por peso, essas informações poderão ser utilizadas para ajustar o *target* a fim de melhorar a consistência da aparência visual dos sorvetes produzidos pelo quiosque da Roboteria, mitigando os efeitos da variação de densidade no volume de produto servido.

Os testes com corpos de prova de tamanho conhecido mostraram que os dados obtidos para a altura dos sorvetes em milímetros apresentam um erro alto, contudo, a margem de erro em torno de dez por cento alcançada é adequada para a aplicação em questão. Já que as medidas serão utilizadas apenas como uma maneira de aprimorar a satisfação dos clientes e a quantidade mínima de sorvete vendido será garantida pelo sistema de dosagem por peso.

6.1 TRABALHOS FUTUROS

Antes que esse sistema possa ser colocado em operação nas unidades da Roboteria, é necessário validá-lo com a utilização de mais corpos de prova para determinar os erros máximos em um conjunto grande de medições. É importante validar também os resultados do sistema de medição, calibrado com amostras conhecidas, em um quiosque em operação, onde as imagens estão sujeitas as variações nas condições de iluminação e a lente da câmera exposta a sujeiras e condensação.

Além disso, a substituição da câmera utilizada por uma de menor campo de visão e maior qualidade de imagem pode melhorar a performance do sistema de detecção dos objetos e aprimorar as medições, devido à redução das distorções causadas pelo campo de visão mais estreito.

É possível, também, estudar a possibilidade de utilizar fontes de iluminação localizada para aprimorar a consistência das cores nas imagens e aprimorar o método de detecção por limiar de cores.

REFERÊNCIAS

- AFONSO, I. **Internet das Coisas: A arquitetura por trás da conectividade**. 2020. Disponível em: <https://zup.com.br/blog/a-arquitetura-da-internet-das-coisas>. Acesso em: 19 nov. 2024.
- ALONSO, V. et al. Industry 4.0 implications in machine vision metrology: an overview. **Procedia Manufacturing**, v. 41, p. 359–366, 2019.
- AVIA SEMICONDUCTOR. **24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales**. 2024. Disponível em: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf. Acesso em: 22 out. 2024.
- BANZI, M. **Primeiros Passos com o Arduino**. 1. ed. [S.l.]: Novatec, 2011. 17-18 p. ISBN 978-85-7522-290-4.
- BARRY, R. **Mastering the FreeRTOS™ Real Time Kernel**. 2016. Disponível em: https://thuviensoc.tnut.edu.vn/bitstream/123456789/1031/1/Mastering_the_FreeRTOS_Real_Time_Kernel.pdf. Acesso em: 28 out. 2024.
- BASSAN, G. **Apresentando novo MCU ESP32-C3 da família Espressif**. 2021. Disponível em: <https://www.makerhero.com/blog/apresentando-novo-mcu-esp32-c3-da-familia-espressif/>. Acesso em: 22 out. 2024.
- BLUM, S. **Cafe X Launches New Robotic Coffee Bar at San Jose International Airport**. 2019. Disponível em: <https://www.businesswire.com/news/home/20191203005389/en/Cafe-X-Launches-New-Robotic-Coffee-Bar-at-San-Jose-International-Airport>. Acesso em: 18 nov. 2024.
- BRADLEY, D. M. et al. Human capital challenges in the food and beverage service industry of canada: finding innovative solutions. **Worldwide Hospitality and Tourism Themes**, v. 9, n. 4, p. 411–423, 2017.
- BRADSKI, G. The OpenCV Library. **Dr. Dobb's Journal of Software Tools**, 2000.
- BRASIL. **Lei nº 8.078, de 11 de setembro de 1990 - Código de Proteção e Defesa do Consumidor**. 1990. Disponível em: <https://www.gov.br/mj/pt-br/assuntos/seus-direitos/consumidor/Anexos/cdc-portugues-2013.pdf>. Acesso em: 9 nov. 2024.
- CASTRO, G. de. **Criando uma Balança com o Arduino**. 2021. Disponível em: <https://www.robocore.net/tutoriais/celula-de-carga-hx711-com-arduino?srsId=AfmBOor9I9izJqkGLR3dfXm5zswW5Kya5bl1I1C6xeRovpPplEhi-wjD>. Acesso em: 7 out. 2024.
- CASTRODALE, J. **Meet Flippy, Chippy, and Hyphen — the Robots That Will Be Making Your Restaurant Meals**. 2023. Disponível em: <https://www.foodandwine.com/white-castle-fry-cooking-robot-flippy-8383969>. Acesso em: 18 nov. 2024.

- DUFFY, K. **Take a look inside a robot fast-food kitchen with fully autonomous ovens, freezers, and cleaning systems that don't require any staff.** 2022. Disponível em: <https://www.businessinsider.com/robot-fast-food-kitchen-autonomous-machines-no-staff-hyper-robotics-2022-2>. Acesso em: 19 nov. 2024.
- FAGO, M.; ROSA, D. **KukavarProxy.** 2019. Disponível em: <https://github.com/lmtsSrl/KUKAVARPROXY>. Acesso em: 24 out. 2024.
- FILHO, C. J. A. **Roboteria e a Economia da Experiência.** 2020. Disponível em: <https://www.roboteria.com.br/post/roboteria-e-a-economia-da-experiencia>. Acesso em: 20 nov. 2024.
- FLINTEC. **O que é uma célula de carga?** 2024. Disponível em: <https://www.flintec.com.br/aprender/celulas-de-carga>. Acesso em: 6 out. 2024.
- GAVLAK, E. **CafeX, a cafeteria em San Francisco onde o barista é um robô.** 2019. Disponível em: <https://cafeinacao.com.br/cafax-a-cafeateria-em-san-francisco-onde-o-barista-e-um-roboto/>. Acesso em: 18 nov. 2024.
- GOFF, H. D.; HARTEL, R. W. Soft-frozen dairy desserts. In: **Ice Cream.** Springer US, 2013. ISBN 978-1-4614-6096-1. Disponível em: https://doi.org/10.1007/978-1-4614-6096-1_8.
- HARTEL, R. W.; RANKIN, S. A.; JR., R. L. B. A 100-year review: Milestones in the development of frozen desserts. **Journal of Dairy Science**, 2017.
- HYPER ROBOTICS. **Perfecting your fast-food.** 2024. Disponível em: <https://www.hyper-robotics.com>. Acesso em: 19 nov. 2024.
- HYPERSKILL. **HSV.** 2024. Disponível em: <https://hyperskill.org/learn/step/13179>. Acesso em: 15 set. 2024.
- IBRAHIM, M. **Uma história dos modelos de visão.** 2024. Disponível em: <https://www.ultralytics.com/pt/blog/a-history-of-vision-models>. Acesso em: 16 nov. 2024.
- IVANOV, S.; WEBSTER, C. Restaurants and robots: public preferences for robot food and beverage services. **Journal of Tourism Futures**, v. 9, n. 2, p. 229–239, abr./mai. 2023.
- KANG, H.-C. et al. HSV Color-Space-Based Automated Object Localization for Robot Grasping without Prior Knowledge. **Applied Sciences**, v. 11, n. 16, 2021. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/11/16/7593>.
- KIRILLOV, A. et al. Segment anything. **arXiv:2304.02643**, 2023. Disponível em: <https://arxiv.org/pdf/2304.02643>.
- LIN, P. **Robots are Coming to the Kitchen What That Could Mean for Society and Culture.** 2024. Disponível em: <https://www.discovermagazine.com/technology/robots-are-coming-to-the-kitchen-what-that-could-mean-for-society-and>. Acesso em: 21 out. 2024.

LIU, X.; DENG, Z.; YANG, Y. Recent progress in semantic image segmentation. **Artificial Intelligence Review**, Springer Science and Business Media LLC, v. 52, n. 2, p. 1089–1106, jun. 2018. ISSN 1573-7462. Disponível em: <https://arxiv.org/abs/1809.10198#>.

MACHADO, P. L. O. et al. Calibração de células de carga utilizando sistema de aquisição de dados de baixo custo. **Revista Brasileira de Física Tecnológica Aplicada**, v. 6, n. 1, p. 28–40, 2019.

O' MAHONY, N. et al. Deep learning vs. traditional computer vision. In: **Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC)**. Springer Nature Switzerland, 2019. p. 128–144. Disponível em: <https://arxiv.org/pdf/1910.13796>.

MAKR SHAKR. **Makr Shagr is first robotic bartending system**. 2015. Disponível em: <https://www.makrshagr.com/media-room/makr-shagr-is-first-robotic-bartending-system>. Acesso em: 18 nov. 2024.

NAYAK, S. **Augmented Reality using ArUco Markers in OpenCV (C++ / Python)**. 2020. Disponível em: <https://learnopencv.com/augmented-reality-using-aruco-markers-in-opencv-c-python/>. Acesso em: 22 nov. 2024.

PETERS, J. F. **Foundations of Computer Vision**. [S.l.]: Springer, 2017. v. 124. 7-12 p. ISBN 978-3-319-30260-7.

RATH, S. **Contour Detection using OpenCV**. 2021. Disponível em: <https://learnopencv.com/contour-detection-using-opencv-python-c/>. Acesso em: 12 nov. 2024.

ROSEBROCK, A. **OpenCV Histogram Equalization and Adaptive Histogram Equalization (CLAHE)**. 2021. Disponível em: <https://pyimagesearch.com/2021/02/01/opencv-histogram-equalization-and-adaptive-histogram-equalization-clahe/>. Acesso em: 27 out. 2024.

ROSEBROCK, A. **OpenCV Morphological Operations**. 2021. Disponível em: <https://pyimagesearch.com/2021/04/28/opencv-morphological-operations/>. Acesso em: 7 out. 2024.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, 1958. Disponível em: <https://www.ling.upenn.edu/courses/cogs501/Rosenblatt1958.pdf>.

SAVOREAT. **The Rise of Restaurant Robotics in the Food Service Industry**. 2022. Disponível em: <https://savoreat.com/the-rise-of-restaurant-robotics-in-the-foodservice-industry/>. Acesso em: 5 out. 2024.

SHU, H. et al. Tinysam: Pushing the envelope for efficient segment anything model. **arXiv preprint arXiv:2312.13789**, 2023.

SIKI, Z.; TAKACS, B. Automatic recognition of aruco codes in land surveying tasks. **Baltic Journal of Modern Computing**, v. 9, 2021.

TAYLOR, C. **O que é a regra do intervalo interquartil?** 2024. Disponível em: <https://www.greelane.com/pt/ciência-tecnologia-matemática/matemática/what-is-the-interquartile-range-rule-3126244>. Acesso em: 7 nov. 2024.

THE ECONOMIST. **The rise of the robochef.** 2018. Disponível em: <https://www.economist.com/science-and-technology/2018/07/12/the-rise-of-the-robochef>. Acesso em: 18 nov. 2024.

THEVAPALAN, A. **Correlation Matrix In Excel: A Complete Guide to Creating and Interpreting.** 2024. Disponível em: <https://www.datacamp.com/tutorial/correlation-matrix-excel>. Acesso em: 17 nov. 2024.

USMANI, M. F. **MQTT Protocol for the IoT.** 2021. Disponível em: https://www.researchgate.net/publication/373640610_MQTT_Protocol_for_the_IoT_-_Review_Paper. Acesso em: 23 out. 2024.

VARGAS, E. **Science Behind Soft Serve Ice Cream Machines.** 2023. Disponível em: <https://freezerplanet.com/blogs/turnkeyinfo/science-behind-soft-serve-ice-cream-machines>. Acesso em: 17 nov. 2024.

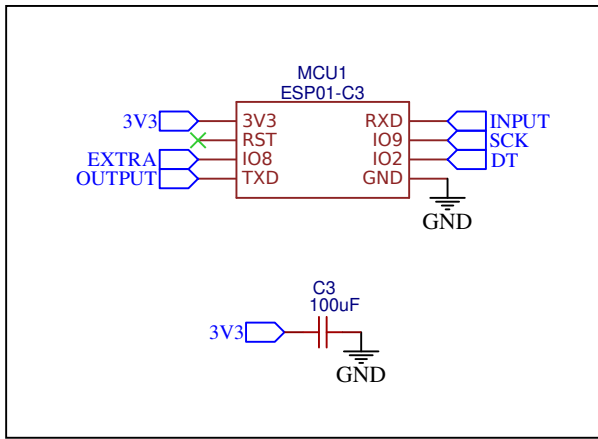
VECCHI, R. de. **Makr Shkr opens robotics bar in Amsterdam.** 2022. Disponível em: <https://globetrender.com/2022/01/13/makr-shkr-opens-robotic-bar-amsterdam/>. Acesso em: 18 nov. 2024.

VIEIRA, P. R. C. **Administração da produção: da revolução industrial à indústria 4.0.** [S.l.]: Autografia, 2021. 6-9 p. ISBN 9786559433803.

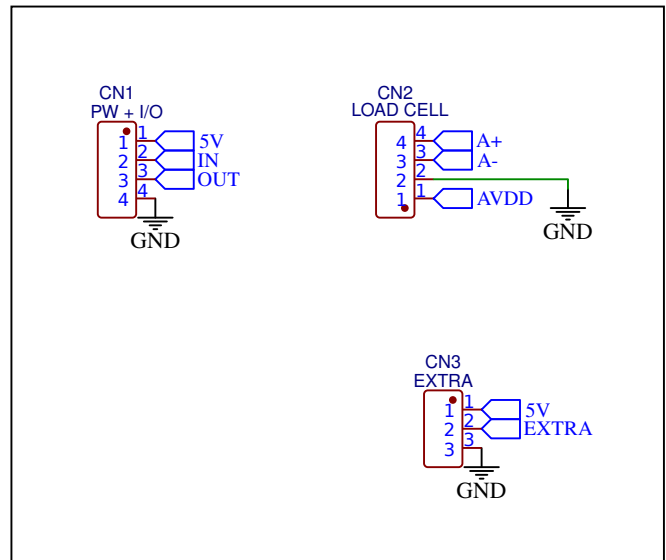
WAGAVKAR, S. **Introduction to the Correlation Matrix.** 2023. Disponível em: <https://www.datacamp.com/tutorial/correlation-matrix-excel>. Acesso em: 18 nov. 2024.

APÊNDICE A - ESQUEMÁTICO DA PCB

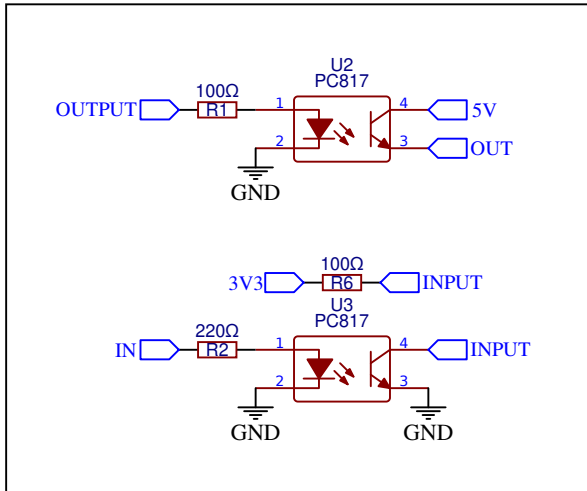
Microcontroller



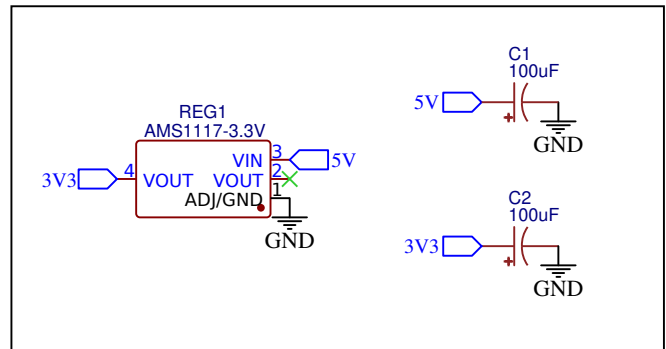
Conectors



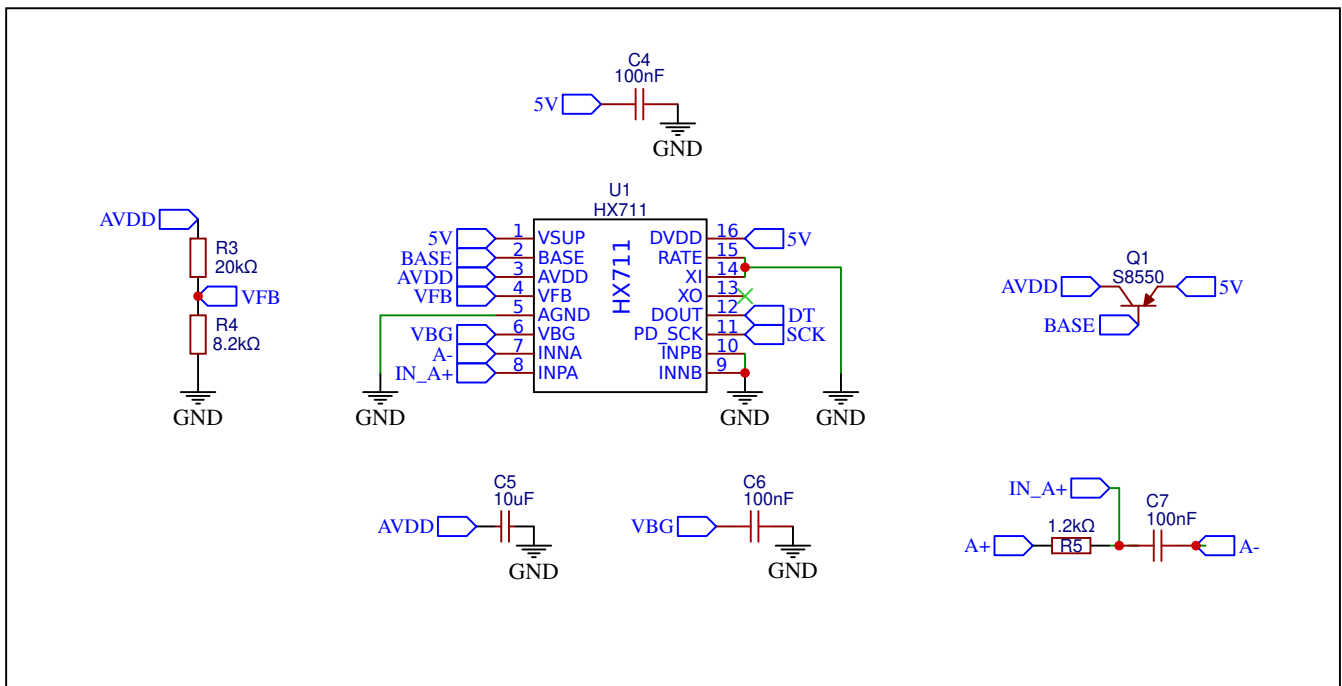
Input and Output



Power



Driver - Load Cell



TITLE:	Load Cell	REV:	2.0	
	Company:	CookingHands Automation	Sheet:	1/1
	Date:	2022-12-11	Drawn By:	Ítalo Coelho

APÊNDICE B - FIRMWARE DA PCB DA CÉLULA DE CARGA

```

//Settings-----
#include "config.h"
//Arduino-----
#include <EEPROM.h>
#include <pwmWrite.h>
//Wifi-----
#include <WiFi.h>
#include <WebServer.h>
#include <HTTPClient.h>
#include <WiFiClient.h>
#include <WiFiManager.h>
#include <PubSubClient.h>
//OTA Update-----
#include <ESPmDNS.h>
#include <WiFiUDP.h>
#include <ArduinoOTA.h>
//Load Cell-----
#include <HX711_ADC.h>
//-----

//Global Variables-----
uint64_t last_command = 0;
bool doTare = false;
bool measure = true;
volatile float reading;
float calibration = -848.78;
bool wifi_connected = false;
bool mqtt_connected = false;
TaskHandle_t LoadCell_handle = nullptr;
//-----

//Objects-----
Pwm servo = Pwm();
HTTPClient http;
WiFiManager wifi;
WiFiClient client;
WebServer server(80);
PubSubClient mqtt(client);
HX711_ADC scale(dataPin, clockPin);
//-----

//Function Prototypes-----
void MQTT_task(void *parameters);
void loadCell_task(void *parameters);
void statusLED_task(void *parameters);
//-----

void setup()
{
  pinMode(IN, INPUT);
  pinMode(OUT, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, HIGH);

  debug_begin(1E+6);

  EEPROM.begin(16);
  if(EEPROM.read(0) != 't')
  {
    EEPROM.write(0, 't');
    EEPROM.put(1, calibration);
    EEPROM.commit();
  }
  EEPROM.get(1, calibration);

  // WiFi Config
  wifi.setHostname(name);
  wifi.setDarkMode(true);
  wifi.setTitle("CookingHands");
#ifdef RELEASE
  wifi.setConnectTimeout(5 * 60);
  wifi.setConfigPortalTimeout(10 * 60);
#endif //RELEASE
  if (!wifi.autoConnect(name))
    ESP.restart();

  // FreeRTOS
  xTaskCreatePinnedToCore( loadCell_task,
                          "Load Cell Handler",
                          8192,
                          nullptr,
                          1,
                          &LoadCell_handle,
                          ARDUINO_RUNNING_CORE );

  xTaskCreatePinnedToCore( MQTT_task,
                          "MQTT Handler",
                          8192,
                          nullptr,
                          1,
                          nullptr,
                          ARDUINO_RUNNING_CORE );
}

```

```

        ARDUINO_RUNNING_CORE );

xTaskCreatePinnedToCore( statusLED_task,
                        "Status LED",
                        1024,
                        nullptr,
                        1,
                        nullptr,
                        ARDUINO_RUNNING_CORE );

// WebServer Callback Functions
server.on("/servo", [] ()
{
    int pos = -1;
    int spd = 180;
    for(int i = 0; i < server.args(); i++)
    {
        String argName = server.argName(i);
        argName.toLowerCase();
        if(argName.indexOf("position")
        pos = server.arg(i).toInt();
        if(argName.indexOf("speed")
        spd = server.arg(i).toInt();
    }
    if(pos >= 0 && pos <= 180 && spd > 0 && spd <= 360)
    {
        server.send(200, "text/plain", "[" + String(name) + "]: OK");
        servo.writeServo(DATA, pos, spd , 1.0);
    }
    else
        server.send(200, "text/plain", "[" + String(name) + "]: Error!\nArgs:\n---\nPosition -> [0, 180] (deg)\nSpeed -> [0, 360] (deg/s)");
});
server.on("/wifi", [] ()
{
    server.send(200, "text/plain", "[" + String(name) + "]: WiFi Settings Reset\nRebooting...");
    wifi.resetSettings();
    delay(100);
    ESP.restart();
});
server.on("/tare", [] ()
{
    scale.tareNoDelay();
    server.send(200, "text/plain", "[" + String(name) + "]: OK");
});
server.on("/start", [] ()
{
    doTare = true;
    measure = true;
    last_command = millis();
    server.send(200);
});
server.on("/stop", [] ()
{
    measure = false;
    server.send(200);
});
server.on("/calibration", [] ()
{
    float weight = server.arg(0).toFloat();
    if (weight > 0)
    {
        vTaskSuspend(LoadCell_handle);
        scale.refreshDataSet();
        calibration = scale.getNewCalibration(weight);
        scale.setCalFactor(calibration);
        vTaskResume(LoadCell_handle);
        debug_println("[HX711]: New Calibration = " + String(calibration));
        EEPROM.put(1, calibration);
        EEPROM.commit();

        server.send(200, "text/plain", "[" + String(name) + "]: Calibrated!\nNew Factor = " + String(calibration));
    }
    else
        server.send(200, "text/plain", "[" + String(name) + "]: Error! -> Known Weight must be larger than 0");
});
server.on("/reading", [] () { server.send(200, "text/plain", "[" + String(name) + "]: " + reading + "g"); });
server.onNotFound([] () { server.send(200, "text/plain", "[" + String(name) + "]: Not Found!"); });
server.begin();

// mDNS Service
MDNS.begin(name);
MDNS.addService("http", "tcp", 80);

// Over-The-Air Update
ArduinoOTA.setHostname(name);
ArduinoOTA.begin();
}

void loop()
{
    if(wifi_connected)
    {
        ArduinoOTA.handle();
    }
}

```



```

    server.handleClient();
}
else
{
    static uint32_t then = millis();
    if(millis() - then >= 1000)
    {
        WiFi.reconnect();
        then = millis();
    }
}
}

void MQTT_task(void *parameters)
{
    TickType_t then;

    mqtt.setKeepAlive(5);
    mqtt.setServer(MQTT_IP, 1883);

    for(;;)
    {
        mqtt_connected = mqtt.connected();
        if(!mqtt_connected)
        {
            if(WiFi.status() != WL_CONNECTED)
            {
                debug_println("[MQTT]: WiFi is Disconnected!");
                vTaskDelay(pdMS_TO_TICKS(100));
                continue;
            }

            debug_print("[MQTT]: Connecting...");
            if(mqtt.connect(name))
            {
                debug_println("OK");
                mqtt.publish("dev/monitor", ("[" + String(name) + "]: Connected!").c_str());
            }
            else
            {
                debug_println("FAIL");
                vTaskDelay(pdMS_TO_TICKS(100));
            }
        }
        else
        {
            mqtt.publish((String(name) + "/reading").c_str(), String(reading).c_str());
            mqtt.loop();

            vTaskDelay(pdMS_TO_TICKS(1000/REFRESHRATE));
        }
    }
}

void loadCell_task(void *parameters)
{
    TickType_t then;

    scale.begin();
    scale.start(2000, true);
    scale.setCalFactor(calibration);
    scale.tare();

    for(;;)
    {
        xTaskDelayUntil(&then, pdMS_TO_TICKS(10));

        if(doTare)
        {
            scale.tare();
            doTare = false;
        }

        if(scale.update())
        {
            reading = scale.getData();

            debug_println("[Load Cell]: " + String(reading) + "g");
        }
    }
}

void statusLED_task(void *parameters)
{
    byte blynks;

    for(;;)
    {
        wifi_connected = WiFi.isConnected();
    }
}

```

```
blynks = 1
+ 1 * wifi_connected
+ 2 * mqtt_connected;

for(int i = 0; i < blynks; i++)
{
  digitalWrite(LED_BUILTIN, HIGH);
  vTaskDelay(pdMS_TO_TICKS(150));
  digitalWrite(LED_BUILTIN, LOW);
  vTaskDelay(pdMS_TO_TICKS(150));
}
vTaskDelay(pdMS_TO_TICKS(1000));
}
```