

# VISÃO COMPUTACIONAL APLICADA À CLASSIFICAÇÃO E SEPARAÇÃO DE PEÇAS COM SISTEMAS EMBARCADOS

## COMPUTER VISION APPLIED TO THE CLASSIFICATION AND SEPARATION OF PARTS USING EMBEDDED SYSTEMS

Wellington de Oliveira Dorta

### RESUMO

Este artigo apresenta o desenvolvimento de dois sistemas automatizados de baixo custo, baseados em visão computacional. O primeiro foi projetado para classificar cápsulas de café, utilizando modelos treinados por meio da plataforma Teachable Machine. O segundo tem como foco a identificação de rótulos em frascos, empregando a técnica de OCR com a biblioteca Pytesseract. Ambos os sistemas foram implementados com um Raspberry Pi 4 e uma Webcam Logitech C270 HD, e integrados a um Arduino Uno responsável pelo controle da esteira, 2 sensores ultrassônicos e outro pelo controle do braço robótico. Os testes realizados demonstraram resultados satisfatórios na triagem visual e na leitura de textos, mesmo em condições de iluminação moderada. Os resultados reforçam o potencial da visão computacional como alternativa eficiente e acessível ao uso de sensores tradicionais em aplicações industriais.

**Palavras-chave:** visão computacional; raspberry pi; OCR; teachable machine; classificação de objetos; automação industrial.

### ABSTRACT

This article presents the development of two low-cost automated systems based on computer vision. The first system was designed to classify coffee capsules using models trained on the Teachable Machine platform. The second focused on identifying product labels through Optical Character Recognition (OCR) with the Pytesseract library. Both systems were implemented using a Raspberry Pi 4 and a Logitech C270 HD webcam, integrated with an Arduino Uno responsible for controlling the conveyor belt, two ultrasonic sensors, and a robotic arm. Experimental tests demonstrated satisfactory performance in both visual sorting and text recognition, even under moderate lighting conditions. The results highlight the potential of computer vision as an efficient and accessible alternative to traditional sensors in industrial applications.

**Keywords:** computer vision; raspberry pi; teachable machine; object classification; industrial automation.

## 1 INTRODUÇÃO

A crescente demanda por sistemas de automação mais inteligentes e adaptáveis tem evidenciado limitações de sensores tradicionais na indústria, que geralmente oferecem apenas dados binários como presença ou ausência, e não são capazes de interpretar características visuais mais complexas. Problemas como falhas na

separação de produtos semelhantes, dificuldade de identificação de embalagens com layout variado, e a ausência de rastreabilidade visual em processos simples ainda são frequentes em linhas produtivas de pequeno e médio porte.

Nesse contexto, a visão computacional surge como uma solução promissora, capaz de interpretar imagens em tempo real e realizar classificações com base em atributos como forma, cor, texto e orientação. Seu uso amplia significativamente a capacidade de decisão de sistemas automatizados e permite o desenvolvimento de soluções mais flexíveis, eficientes e acessíveis.

### **1.1 Problema de pesquisa**

Como aplicar visão computacional, utilizando sistemas embarcados de baixo custo, para automatizar tarefas de classificação e separação de produtos em ambientes industriais, superando as limitações dos sensores convencionais em termos de reconhecimento visual, adaptabilidade e custo-benefício?

### **1.2 Objetivo(s)**

Desenvolver dois experimentos de sistemas automatizados com visão computacional embarcada, utilizando um Raspberry Pi 4 e Webcam Logitech C270 HD. O primeiro experimento explora a classificação de objetos inteiros — como cápsulas de café — com modelos treinados no Teachable Machine, ilustrando a possibilidade de aplicação em praticamente qualquer tipo de produto. O segundo experimento demonstra a leitura e identificação de rótulos utilizando OCR com a biblioteca Pytesseract, uma abordagem mais leve e eficiente para sistemas que trabalham com produtos rotulados.

### **1.3 Justificativa**

Ao replicar processos visuais com boa precisão e baixo custo, a visão computacional se mostra uma alternativa viável para modernizar linhas de produção, reduzindo a dependência de sensores dedicados que muitas vezes tem um custo elevado. Os experimentos realizados neste trabalho reforçam o potencial dessa tecnologia como prova de conceito, demonstrando sua aplicação em diferentes cenários, com flexibilidade, escalabilidade e compatibilidade com hardware acessível. A adoção de soluções como essa tende a democratizar o acesso à automação inteligente, especialmente entre pequenas e médias empresas, contribuindo para a inovação e aumento da competitividade no setor industrial.

## **2 REVISÃO DE LITERATURA**

A visão computacional, ramo da inteligência artificial voltado à interpretação de imagens e vídeos por máquinas, permite que sistemas tomem decisões com base em padrões visuais. De acordo com Szeliski (2010), essa área abrange desde a aquisição até o processamento de imagens e a extração de informações relevantes para a automação.

Com o avanço de sistemas embarcados e dos mini computadores de placa única (SoC) como o Raspberry Pi 4 usado nesse projeto, tornou-se possível desenvolver soluções acessíveis com processamento local. Nesse cenário, a biblioteca OpenCV destaca-se por oferecer recursos robustos de filtragem, segmentação e pré-processamento de imagens (BRADSKI; KAEHLER, 2017).

Entre as ferramentas modernas, o Teachable Machine — desenvolvido pelo Google — permite treinar classificadores visuais personalizados com base em redes neurais convolucionais, mesmo sem conhecimento em programação (LECUN; BENGIO; HINTON, 2015; GIBSON et al., 2020).

Outra técnica relevante é o OCR (Reconhecimento Óptico de Caracteres), utilizado para extrair textos de imagens. A biblioteca Pytesseract, interface Python para o Tesseract OCR, tem se mostrado eficaz na leitura de rótulos e códigos, especialmente quando combinada com técnicas de binarização e remoção de ruído (SMITH, 2007).

Grandes empresas como Tesla, Nestlé e Samsung já aplicam essa tecnologia em inspeções, controle de montagem e verificação de embalagens. Além disso, estudos apontam sua expansão para áreas como saúde, segurança e mobilidade urbana, mostrando o potencial da visão computacional como solução acessível e versátil para diferentes setores industriais (GIBSON et al., 2020; KATZ, 2020).

### 3 METODOLOGIA

Esta pesquisa é classificada como experimental, com base bibliográfica e documental, fundamentada no estudo de ferramentas de visão computacional, aprendizado de máquina e sistemas embarcados. O objetivo principal foi desenvolver um sistema de baixo custo para demonstração da classificação e separação automatizada de objetos por meio de visão computacional embarcada.

A metodologia foi estruturada em cinco etapas: levantamento do material e ferramentas, planejamento do sistema, aquisição de dados, treinamento dos modelos e implementação prática dos protótipos.

#### 3.1 Materiais e ferramentas

Para o desenvolvimento do sistema foram utilizados componentes de fácil integração, com ênfase em baixo custo e compatibilidade com aplicações embarcadas:

- **Raspberry Pi 4 Model B** como unidade de processamento;
- **Câmera Logitech C270 HD** para captura das imagens na esteira;
- **Arduino Uno R3 (2 unidades)**: um para o controle da esteira e outro para o braço robótico;
- **Sensores ultrassônicos HC-SR04** para detecção de presença antes e após a leitura;
- **Esteira motorizada em escala reduzida**, com estrutura física simples;
- **Braço robótico com servos SG90**, acionado por driver externo;
- **Micro switch** para controle de fim de curso;
- **Protoboard, jumpers e cabos diversos** para montagem dos circuitos;
- **Fonte de alimentação 5V/3A** para o Raspberry Pi.

No software, foram adotadas:

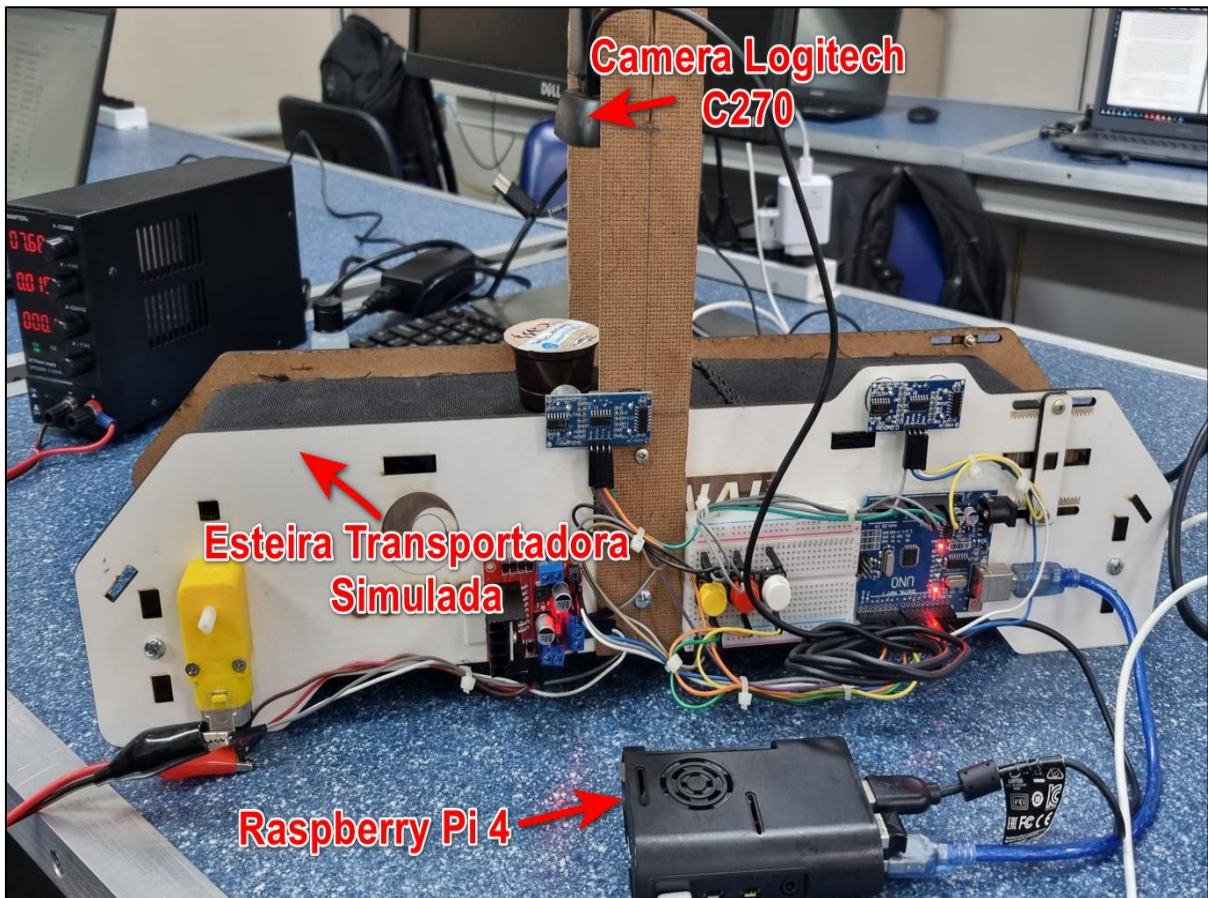
- **OpenCV (Python)** para pré-processamento das imagens;
- **Teachable Machine (Google)** para treinamento do modelo de classificação no experimento 1;
- **Tesseract OCR com Pytesseract** para leitura de rótulos no experimento 2;

- NumPy, Keras e TensorFlow para suporte ao modelo visual;
- PyCharm e Arduino IDE como ambientes de desenvolvimento.

### 3.2 Planejamento do Sistema

O sistema foi construído com foco na separação automatizada de objetos, utilizando uma mini esteira controlada por Arduino Uno e uma câmera conectada ao Raspberry Pi 4 para captura e processamento de imagem. A estrutura geral do protótipo pode ser observada na Figura 1.

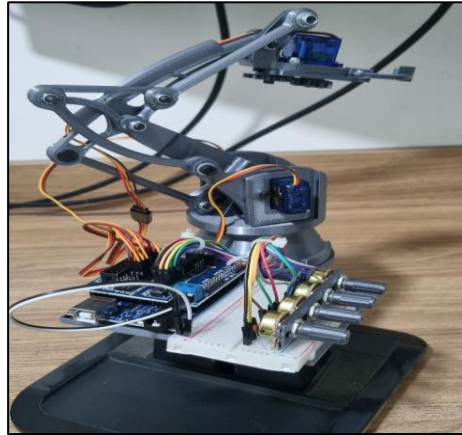
**Figura 1** – Visão geral do sistema embarcado com câmera Logitech C270 e Raspberry Pi 4.



**Fonte:** Elaborado pelo próprio autor (2025).

Na prática, os objetos percorrem a esteira e, ao entrarem no campo de visão da câmera, um sensor ultrassônico envia sinal ao Arduino para pausar o motor, garantindo a captura adequada da imagem. Após o reconhecimento, a esteira é reativada e, ao final do trajeto, um segundo sensor aciona o braço robótico. A separação física é feita conforme o resultado da classificação. O componente responsável por essa atuação é ilustrado na Figura 2.

**Figura 2** - Braço robótico utilizado no sistema



**Fonte:** Elaborado pelo próprio autor (2025).

Foram realizados dois experimentos: o primeiro utilizando cápsulas de café classificadas por modelo treinado no Teachable Machine (Figura 3), e o segundo com frascos de produtos cosméticos identificados via OCR (Figura 4).

**Figura 3** – Cápsulas de café utilizadas no experimento com Teachable



**Fonte:** Elaborado pelo próprio autor (2025).

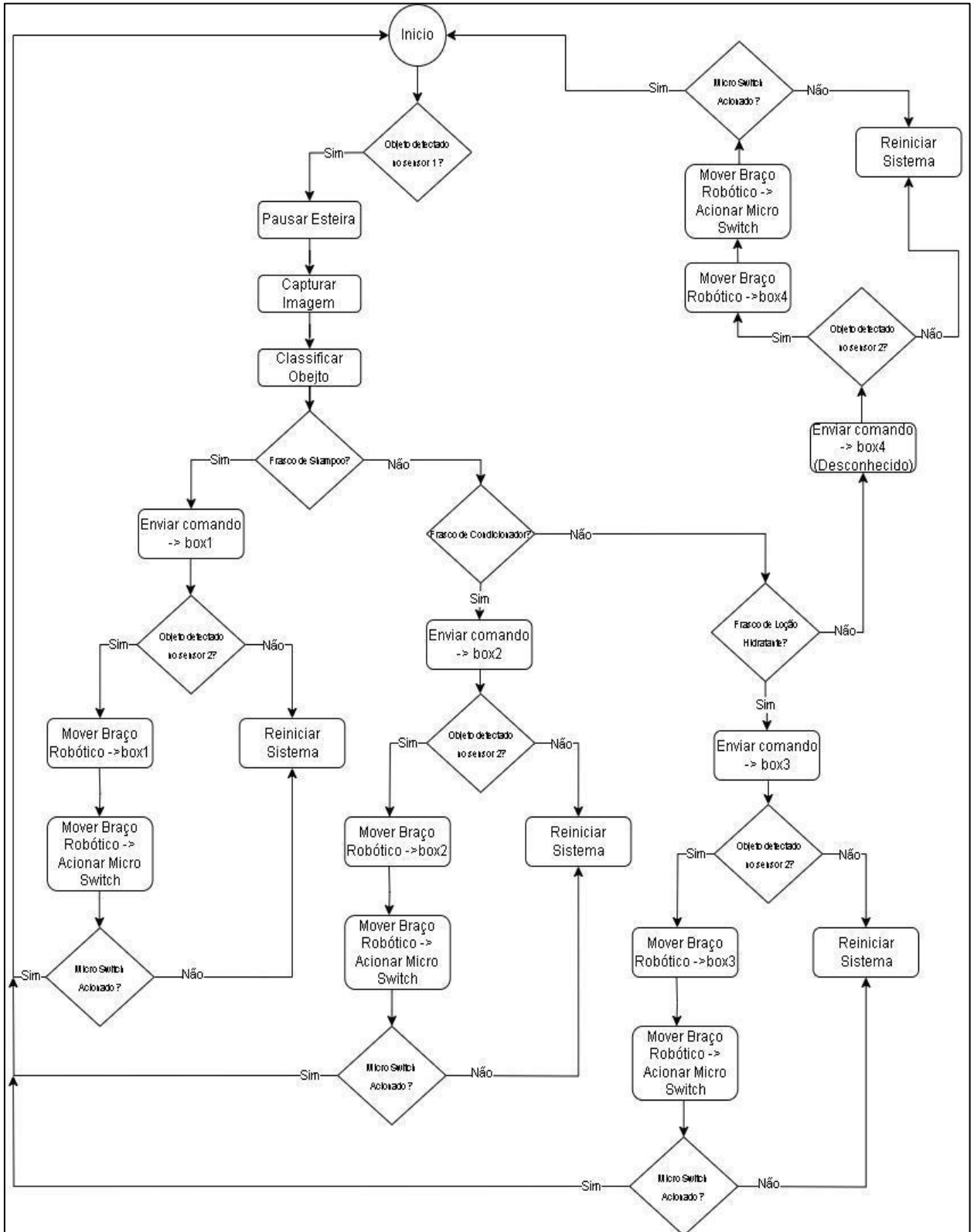
**Figura 4** – Frascos de shampoo, condicionador e loção hidratante utilizados no experimento com OCR.



**Fonte:** Elaborado pelo próprio autor (2025).

A lógica completa do sistema, desde a detecção até a separação, está representada no fluxograma da Figura 5.

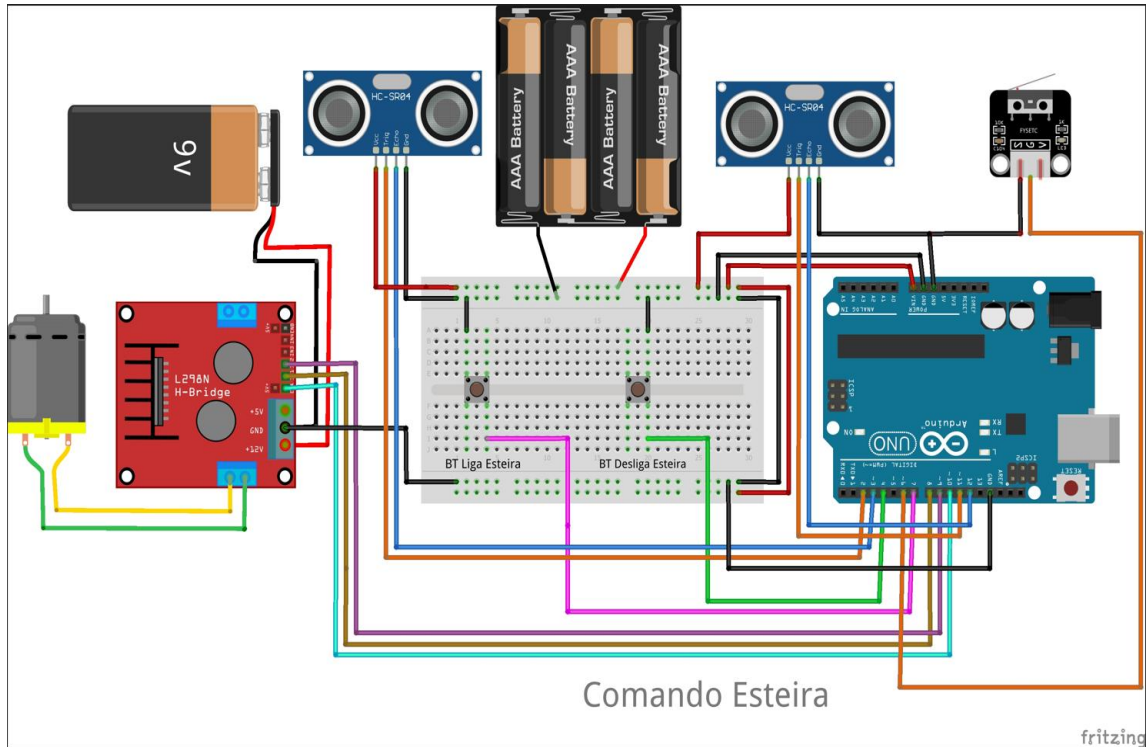
**Figura 5** – Fluxograma de funcionamento do sistema de separação automatizada.



Fonte: Elaborado pelo próprio autor (2025).

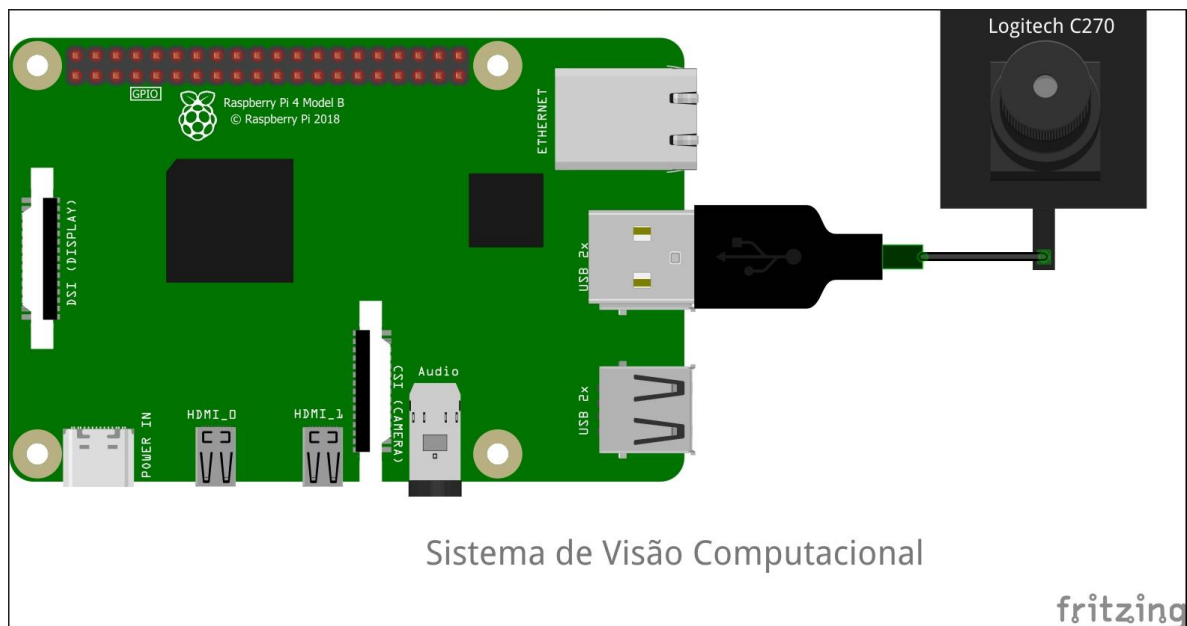
Além disso, foram elaborados três esquemas de ligação elétrica. A Figura 6 apresenta o circuito da esteira e sensores, a Figura 7 mostra a integração entre o Raspberry Pi e a câmera, e a Figura 8 detalha o controle do braço robótico.

**Figura 6** – Esquema elétrico da esteira com sensores e motor.



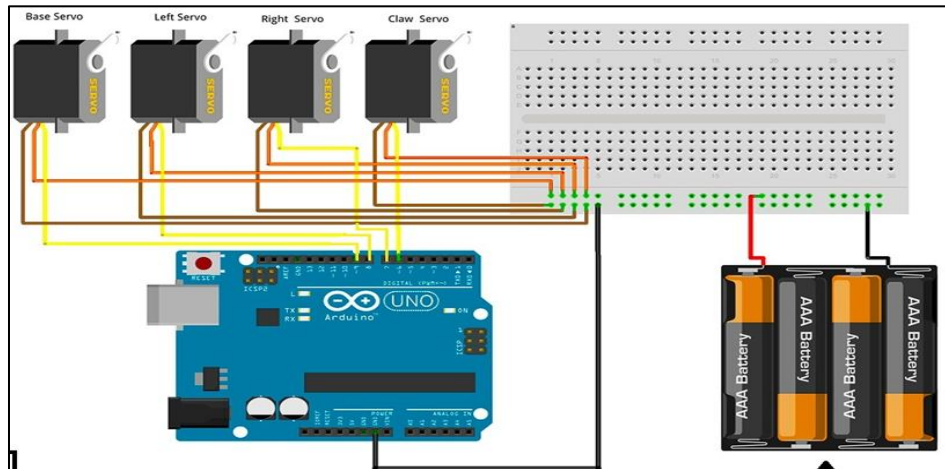
Fonte: Elaborado pelo próprio autor (2025).

**Figura 7** – Esquema do sistema de visão computacional com Raspberry Pi.



Fonte: Elaborado pelo próprio autor (2025).

**Figura 8** – Esquema de ligação do braço robótico com Arduino Uno.



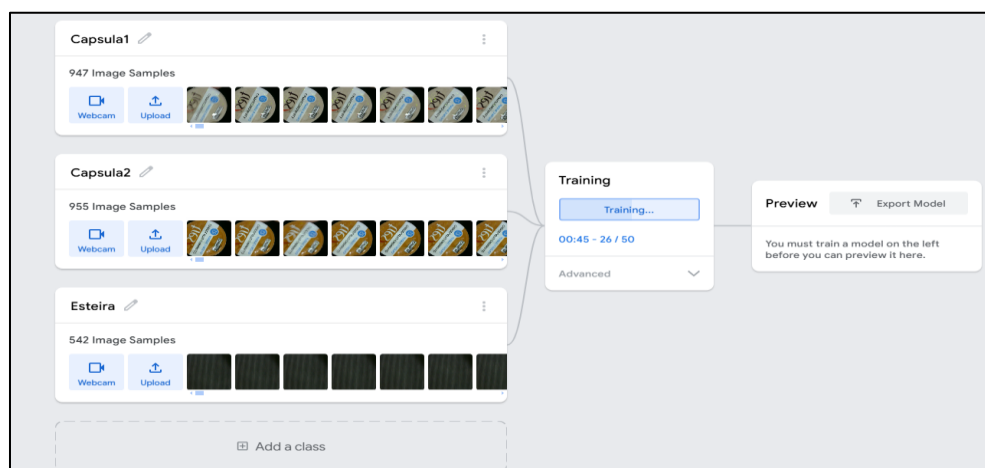
Fonte: Elaborado pelo próprio autor (2025).

### 3.3 Aquisição de Dados e Treinamento

Para o primeiro experimento, voltado à classificação visual de cápsulas de café, foi utilizada a plataforma Teachable Machine do Google. A ferramenta, baseada em redes neurais convolucionais (CNNs), permite treinar modelos diretamente pelo navegador, sendo ideal para aplicações rápidas e educativas sem necessidade de programação.

Durante o treinamento, foram coletadas cerca de 1000 imagens para cada classe de cápsula (por exemplo, azul e preta), além de imagens do fundo da esteira com diferentes condições de iluminação, ângulo e posição, com o objetivo de aumentar a capacidade de generalização do modelo.

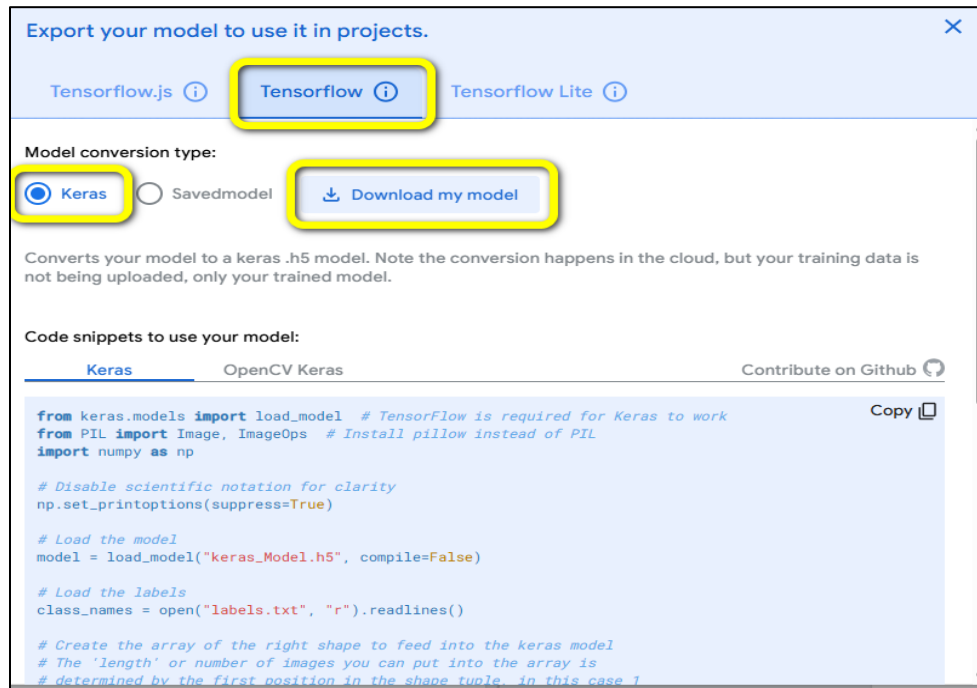
**Figura 9** – Realização do treinamento no Teachable Machine.



Fonte: Elaborado pelo próprio autor (2025).

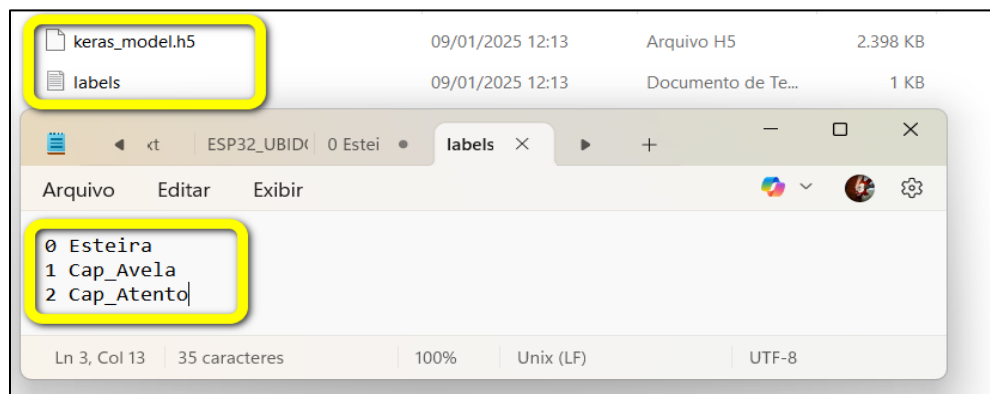
Após o treinamento, o modelo foi exportado em formato compatível com TensorFlow, contendo arquivos como `keras_model.h5` e `labels.txt`, que foram carregados no Raspberry Pi com auxílio das bibliotecas TensorFlow, Keras, NumPy e OpenCV.

Figura 10 – Formato de exportação do modelo treinado.



Fonte: Elaborado pelo próprio autor (2025).

Figura 11 – Arquivos gerados após exportação do modelo.



Fonte: Elaborado pelo próprio autor (2025).

O modelo foi integrado ao sistema embarcado e passou a classificar os objetos em tempo real. A inferência foi realizada localmente no Raspberry Pi 4, com a imagem pré-processada alimentando a rede, e a classe identificada sendo usada para acionar o braço robótico por meio de comandos enviados ao Arduino.

### 3.4 Integração com o Sistema

O código foi implementado em Python, executado no Raspberry Pi 4, com as bibliotecas **TensorFlow**, **Keras**, **NumPy**, **OpenCV** e **Serial**. Esse conjunto permitiu o reconhecimento visual em tempo real e o controle físico do sistema de separação.

As etapas principais do processo foram:

## 1. Importação e configuração:

Foram utilizadas bibliotecas para manipulação de imagem, execução de modelo treinado e envio de comandos ao Arduino. A comunicação serial foi estabelecida com taxa de 9600 bps.

## 2. Carregamento do modelo:

O modelo treinado no Teachable Machine foi carregado com `load_model()`, incluindo suporte à camada `DepthwiseConv2D`.

## 3. Captura e pré-processamento da imagem:

A imagem da câmera foi redimensionada para 224x224 pixels, normalizada e suavizada com filtro Gaussiano. Essa etapa garantiu a qualidade da entrada para a rede neural.

## 4. Inferência e tomada de decisão:

A imagem processada foi enviada ao modelo, que retornou as probabilidades de classificação. Um limiar de confiança de 80% foi adotado. A classe com maior probabilidade definia o comando enviado ao Arduino para acionar o braço robótico.

Figura 12 – Envio de comandos ao Arduino com base na classificação obtida.

```
# Mostrar resultado com limiar de confiança
if confidence > 0.8: # Ajuste o limiar conforme necessário
    label = f"{classes[smoothed_index]}: {confidence:.2f}"
    color = (0, 255, 0)

    # Enviar comando ao Arduino com base na classe detectada
    if smoothed_index == 1: # Cap_Avela
        enviar_comando('L') # Envia comando para a caixa da esquerda
    elif smoothed_index == 2: # Cap_Atento
        enviar_comando('R') # Envia comando para a caixa da direita
    else:
        # Classificar como "Desconhecido" se a confiança for baixa
        label = "Desconhecido"
        color = (0, 0, 255)
```

Fonte: Elaborado pelo próprio autor (2025).

## 5. Interface visual e encerramento:

A imagem com a classificação e confiança era exibida em tempo real na tela. O sistema era finalizado com a tecla 'q', liberando os recursos de câmera e comunicação.

### 3.5 Reconhecimento Óptico de Caracteres (OCR)

No segundo experimento, foi utilizada a biblioteca **Tesseract OCR**, integrada ao Python via **Pytesseract**, com o objetivo de identificar produtos por meio da leitura dos rótulos presentes nos frascos de shampoo, condicionador e loção hidratante. A

imagem era capturada no momento em que o sensor ultrassônico detectava a presença do objeto na região de leitura. A câmera Logitech C270 enviava a imagem ao Raspberry Pi, onde era processada com técnicas de ajuste de brilho, contraste, conversão para tons de cinza, binarização adaptativa e remoção de ruídos (OpenCV), a fim de otimizar o reconhecimento dos caracteres. Após o pré-processamento, o OCR extraía as palavras da imagem. A Figura 13 mostra os fragmentos de extração que foram definidos para identificação dos frascos, que foram definidos da seguinte forma:

- Shampoo: ['sham', 'ham', 'amp', 'mpo', 'poo', 'hoo', 'shampoo']
- Condicionador: ['con', 'ond', 'ndi', 'dic', 'icio', 'ion', 'nado', 'ador', 'condicionador']
- Loção Hidratante: ['loc', 'oca', 'cao', 'ção', 'hidra', 'idr', 'rat', 'atina', 'hidratante', 'loção']

**Figura 13** – Fragmentos de identificação.

```
# Fragmentos de identificação
fragmentos_shampoo = ['sham', 'ham', 'amp', 'mpo', 'poo', 'hoo', 'shampoo']
fragmentos_condicionador = ['con', 'ond', 'ndi', 'dic', 'icio', 'ion', 'nado', 'ador', 'condicionador']
fragmentos_locao = ['loc', 'oca', 'cao', 'ção', 'hidra', 'idr', 'rat', 'atina', 'hidratante', 'loção']

def contem_fragmentos(lista_fragmentos, texto):
    return sum(1 for frag in lista_fragmentos if frag in texto)
```

**Fonte:** Elaborado pelo próprio autor (2025).

Uma vez identificada a categoria do produto, o sistema avalia se o novo resultado difere da última predição e se respeitará um intervalo mínimo de 3 segundos entre as decisões, evitando múltiplas detecções do mesmo item. Se os critérios forem atendidos, um comando é enviado ao Arduino responsável pelo braço robótico por meio da comunicação serial (porta COM11), a lógica é basicamente essa abaixo e como é mostrado no código da Figura 14.

- "box1" → Frascos de shampoo
- "box2" → Frascos de condicionador
- "box3" → Frascos de loção hidratante

A lógica de controle armazena o comando de separação até que o sensor no final da esteira (Arduino na COM13) detecte a chegada do produto, enviando então o sinal "ready" para acionar o braço robótico de forma sincronizada.

**Figura 14** – Lógica dos comandos de envio após identificação via OCR.

```
if resultado in ["Shampoo", "Condicionador", "Locao Hidratante"] and resultado != ultimo_resultado and \
    (tempo_atual - ultimo_envio > intervalo_envio) and braco_disponivel:
    if resultado == "Shampoo":
        enviar_braco("box1\n")
        print("➡ Enviado comando: box1 (Shampoo)")
    elif resultado == "Condicionador":
        enviar_braco("box2\n")
        print("➡ Enviado comando: box2 (Condicionador)")
    elif resultado == "Locao Hidratante":
        enviar_braco("box3\n")
        print("➡ Enviado comando: box3 (Loção)")

    comando_enviado = True
    ultimo_resultado = resultado
    ultimo_envio = tempo_atual
```

**Fonte:** Elaborado pelo próprio autor (2025).

## 4 RESULTADOS E DISCUSSÕES

Esta seção apresenta os principais resultados obtidos com a aplicação prática dos dois experimentos em um ambiente simulado. Utilizando sistemas embarcados de baixo custo, buscou-se avaliar a viabilidade da visão computacional como alternativa eficiente para automatizar tarefas de classificação e separação de objetos.

### 4.1 Experimento 1 – Classificação Visual com Teachable Machine

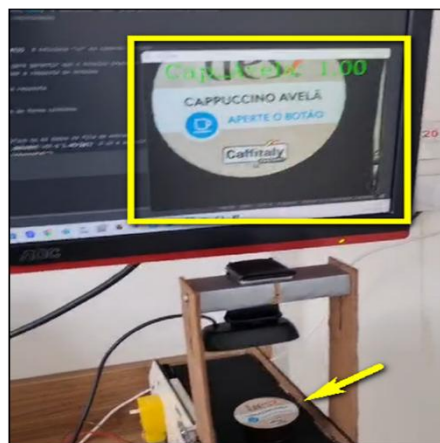
O primeiro experimento envolveu a classificação de cápsulas de café com um modelo treinado na plataforma Teachable Machine. As cápsulas eram transportadas por uma esteira, e a identificação visual era feita em tempo real, acionando o braço robótico para realizar a separação física.

**Figura 15** – Reconhecimento da cápsula Espresso Atento com 100% de confiabilidade.



**Fonte:** Elaborado pelo próprio autor (2025).

**Figura 16** – Reconhecimento da cápsula Cappuccino Avelã com 100% de confiabilidade.



**Fonte:** Elaborado pelo próprio autor (2025).

A precisão observada foi superior a 90%, com falhas ocasionais devido a iluminação inadequada ou desalinhamento das cápsulas.

**Tabela 1** – Resultados da classificação automática com Teachable Machine.

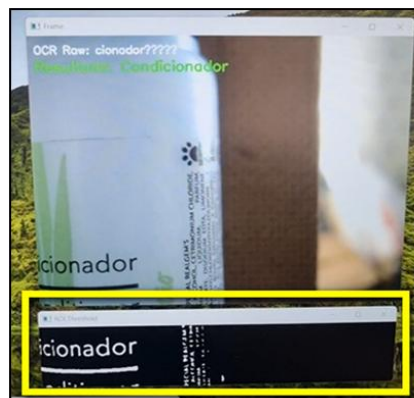
Classe real	Total testado	Classificações corretas	Precisão (%)
Expresso Atento	15	14	93,30%
Cappuccino Avelã	15	13	86,60%

O tempo médio de resposta entre captura e decisão foi inferior a 1 segundo, demonstrando que o sistema é aplicável em processos industriais com velocidade controlada.

#### 4.2 Experimento 2 – Leitura de Rótulos com OCR (Pytesseract)

Neste experimento, frascos de shampoo, condicionador e loção hidratante foram identificados pela leitura de seus rótulos, utilizando OCR com a biblioteca Pytesseract. O sistema aplicou pré-processamento na região inferior da imagem (ROI), com binarização e suavização via OpenCV para melhorar a precisão da leitura.

**Figura 17** – Pré-processamento da imagem e extração de texto com Pytesseract.



**Fonte:** Elaborado pelo próprio autor (2025).

A identificação foi feita pela busca de fragmentos-chave, como "sham", "cond" ou "hidr", atribuindo a categoria com maior ocorrência. As decisões foram convertidas em comandos seriais enviados ao Arduino para ativar o braço robótico.

**Figura 17** – Braço robótico coletando o frasco da esteira para envio à caixa de destino.



**Fonte:** Elaborado pelo próprio autor (2025).

Para simular uma peça fora do padrão, uma etiqueta com a palavra “sabão” foi fixada em um frasco. O sistema corretamente rejeitou o item, direcionando-o à posição de descarte.

**Figura 18** – Simulação de leitura de peça desconhecida com etiqueta “sabão”.



Fonte: Elaborado pelo próprio autor (2025).

**Figura 19** – Peça desconhecida sendo levada a posição de descarte.



Fonte: Elaborado pelo próprio autor (2025).

**Tabela 2** – Resultados da classificação OCR com Pytesseract e separação física dos frascos.

Classe esperada	Total testado	Classificações corretas	Precisão (%)
Shampoo	10	9	90%
Condicionador	10	9	90%
Loção Hidratante	10	8	80%
Peça desconhecida	3	3	100%

### 4.3 Pontos de Atenção e Dificuldades Enfrentadas

Durante os testes, foram enfrentados desafios como variações de iluminação, enquadramento dos objetos e ruídos visuais que afetam a acurácia da classificação e leitura via OCR. Esses problemas foram contornados com o uso de iluminação artificial, definição de uma ROI fixa e aplicação de filtros no OpenCV.

Sensores ultrassônicos apresentaram falhas em velocidades elevadas, resolvidas com ajuste da velocidade da esteira. No braço robótico, interferências nos potenciômetros exigiram o uso de capacitores e calibração por software.

Por fim, a sincronização entre os módulos foi otimizada com travas de execução e controle de fluxo, assegurando a atuação correta do sistema. Todas as dificuldades foram superadas com soluções práticas e de baixo custo, viabilizando o funcionamento confiável do protótipo.

## 5 CONCLUSÃO

Os experimentos realizados demonstraram a viabilidade do uso da visão computacional com sistemas embarcados de baixo custo em aplicações de automação industrial. Tanto a abordagem por classificação visual com Teachable Machine quanto o reconhecimento de texto via OCR mostraram resultados satisfatórios, com índices de acerto superiores a 90%.

Apesar das limitações enfrentadas, como sensibilidade à iluminação e calibração de componentes, o sistema apresentou desempenho estável e replicável. A solução proposta se destaca pela acessibilidade, baixo custo e potencial de adaptação para diversos tipos de produtos e ambientes industriais.

## REFERÊNCIAS

- BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- GOOGLE. *Teachable Machine*. Disponível em: <https://teachablemachine.withgoogle.com/>. Acesso em: 10 jul. 2025.
- SMITH, R. An Overview of the Tesseract OCR Engine. *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)*, 2007.
- MACHADO, A. S.; ZUFFO, M. K. Visão computacional: conceitos e aplicações. *Revista da Escola de Engenharia da UFMG*, v. 7, n. 2, p. 29–37, 2014.
- MARTINS, J. P.; MANO, W. L. Sistemas embarcados: conceitos, tecnologia e aplicações. *Revista Tecnologia e Sociedade*, v. 15, n. 33, p. 144–151, 2019.

## APÊNDICE A – REPOSITÓRIO DO PROJETO

O código-fonte, diagramas e arquivos do projeto estão disponíveis em:

 <https://github.com/tomtecsolutions/visao-embarcada>



## AGRADECIMENTOS

Agradeço ao SENAI Roberto Mange pela oportunidade de desenvolver este projeto, bem como aos colegas e professores — em especial ao professor Michel Chaparro — que contribuíram com sugestões técnicas e apoio durante a execução dos experimentos.

## SOBRE O(S)AUTOR(ES)

Sobre os autores:

### <sup>i</sup> Wellington de Oliveira Dorta (Autor 1)



Graduado em Engenharia Eletrônica pela Faccamp (2018), cursa Pós-Graduação em Automação Industrial no SENAI Roberto Mange. Atua como Engenheiro de Campo Jr. na Samsung, com foco em falhas de áudio e vídeo, e tem experiência com C, C++ e Python aplicadas a sistemas embarcados, visão computacional e automação industrial.